The Dissertation Committee for Xiao Pu Certifies that this is the approved version of the following dissertation:

A BANDWIDTH-ENHANCED FRACTIONAL-N PLL THROUGH REFERENCE MULTIPLICATION

Committee:

Jacob Abraham, Supervisor

Axel Thomsen, Co-Supervisor

Arjang Hassibi

Michael Orshansky

David Pan

Lorenzo Alvisi

A BANDWIDTH-ENHANCED FRACTIONAL-N PLL THROUGH REFERENCE MULTIPLICATION

by

Xiao Pu, B.S.; M.S.; M.S.E.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2011

Dedication

I dedicate this dissertation to my parents, my husband and my daughter for their unwavering support.

Acknowledgements

I would like to thank Professor Jacob Abraham for accepting me as his student, including me in his vibrant research group, teaching me by example how to do research and for always being there for me whenever I had difficulties to overcome. I will treasure memories of time I spent in his research group and am indebted to him for all his support. I thank Dr. Axel Thomsen for introducing me to the world of PLL design. Dr. Thomsen got me excited about analog design; he also freely shared his ideas with me on various approaches he was considering for improving PLL performance. I will cherish the numerous luncheons we shared where designs and solutions were sketched on napkins and any available paper. Many of those ideas form the main body of work in this dissertation. I also thank Professor Arjang Hassibi for being a source of support for me during some of the challenging times I faced during this work. I learnt a lot from the Integrated Sensors class that Professor Hassibi taught and was able to use it in my research. Thank you Professor Hassibi, for serving on my committee. I would also like to thank Professor David Pan, Professor Michael Orshansky and Professor Alvisi Lorenzo for taking an interest in my work, guiding through the qualifying process and for serving on my committee.

At Texas Instruments, I would like to thank my manager Dr. Krishnaswamy Nagaraj for all his support. This dissertation could not have been completed without his constant encouragement and insightful guidance. The passion that Dr. Nagaraj has for research constantly inspires me. Dr. Nagaraj was also instrumental in arranging for the chip to be fabricated by TI. I thank my colleague and mentor Dr. Ajay Kumar for always being available to listen to my problems and for suggesting possible solutions. I learnt a great deal from our discussions. I thank Yunfei Tuan and Kiran Damle for doing the

layout of the chip described in this dissertation. I thank Dan Edmondson, Dr. Joonsung Park, Yuwen Zou and Srinadh Madhavapeddi for helping me with the synthesis of the digital block, and Ayaskanta Behera for the placement and routing for this block.

I benefitted enormously from the time I spent at the University of Texas, ECE department. I am indebted to Professor Ranjit Gharpurey for teaching me how research is done. I learnt many aspects of IC design from Professor Gharpurey's RFIC class. Prof Gharpurey took an interest in my work and patiently advised me on many aspects related to my work. I also thank Adjunct Professor Eric Swanson for teaching me analog design and for helping me, both during the times I spent in academia and industry.

Bandwidth-enhanced Fractional-N PLL through Reference Multiplication

Publication No._____

Xiao Pu, Ph.D.

The University of Texas at Austin, 2011

Supervisors: Jacob Abraham and Axel Thomsen

The loop bandwidth of a fractional-N PLL is a desirable parameter for many applications. A wide bandwidth allows a significant attenuation of phase noise arising from the VCO. A good VCO typically requires a high Q LC oscillator. It is difficult to build an on-chip inductor with a high Q factor. In addition, a good VCO also requires a lot of power. Both these design challenges are relaxed with a wide loop bandwidth PLL. However a wide loop bandwidth reduces the effective oversampling ratio (OSR) between the update rate and loop bandwidth and makes quantization noise from the ΔΣ modulator a much bigger noise contributor. A wide band loop also makes the noise and linearity performance of the phase detector more significant. The key to successful implementation of a wideband fractional-N synthesizer is in managing jitter and spurious performance. In this dissertation we present a new PLL architecture for bandwidth extension or phase noise reduction. By using clock squaring buffers with built-in offsets, multiple clock edges are extracted from a single cycle of a sinusoidal reference and used for phase updates, effectively forming a reference frequency multiplier. A higher update

rate enables a higher OSR which allows for better quantization noise shaping and makes a wideband fractional-N PLL possible. However since the proposed reference multiplier utilizes the magnitude information from a sinusoidal reference to obtain phases, the derived new edges tend to cluster around the zero-crossings and form an irregular clock. This presents a challenge in lock acquisition. We have demonstrated for the first time that an irregular clock can be used to lock a PLL. The irregularity of the reference clock is taken into account in the divider by adding a cyclic divide pattern along with the ΔΣ control bits, this forces the loop to locally match the incoming patterns and achieve lock. Theoretically this new architecture allows for a 6x increase in loop BW or a 24dB improvement in phase noise. One potential issue associated with the proposed approach is the degraded spurious performance due to PVT variations, which lead to unintended mismatches between the irregular period and the divider pattern. A calibration scheme was invented to overcome this issue. In simulation, the calibration scheme was shown to lower the spurs down to inherent spurs level, of which the total energy is much less than the integrated phase noise. A test chip for proof of concept is presented and measurements are carefully analyzed.

Table of Contents

List of Tables

List of Figures

Chapter 1: Overview

1.1 PLL IN A GLANCE

Phase lock loops (PLLs) are widely used for frequency synthesis and clock generation in modern Systems-On-Chip. A PLL is a control system which uses negative feedback to align the clock phase of a voltage controlled oscillator (VCO) to that of the input reference. Typically a Quartz Crystal Oscillator (XO) is used to generate the input reference. A quartz crystal is a piezo-electric device which produces a mechanical oscillation when a voltage source is applied; the frequency of oscillation is determined by the shape, the cut and the elastic constants of the crystal [1]. Each XO provides a precise, fixed frequency source often in the range of a few kHz to tens of MHz. A higher oscillation frequency is also possible by tuning a crystal to its harmonics, called overtone frequencies. An XO exhibits extremely low phase noise due to its high Q factor, which is several orders of magnitude higher than an LC tank oscillator [23]. An XO also offers superior thermal stability often within 100ppms. However an XO is typically designed around a few standard frequencies. The main advantage of using a PLL for frequency multiplication lies in its ability to synthesize a highly tunable clock source using a local VCO with its in-band phase noise performance approaching that of the XO.

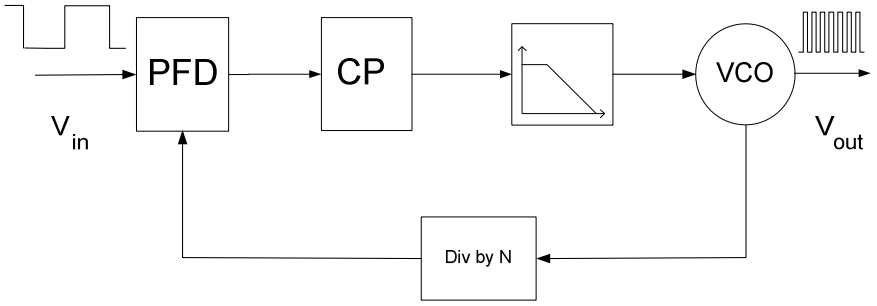


Figure 1.1: A Typical PLL

A block diagram of a typical PLL block diagram is shown in Fig. 1.1. A phase frequency detector (PFD) or a phase detector (PD) is used to compare the phase between the reference clock and the feedback clock; the phase difference is measured using a charge pump (CP); the charge pump current is then converted to voltage across the loop filter (LPF) capacitors; the voltage signal gets further smoothed out in the LPF and is used to tune the VCO; a divided-down version of the VCO clock feeds back to the PD and the loop is closed. In steady state frequency multiplication is achieved, $f_{vco} = N \cdot f_{ref}$, where N is the divide ratio. In this equation N can either be an integer or an integer plus a fraction, giving rise to the terms integer-N PLL and fractional-N PLL, respectively.

1.2 CHALLENGES IN PLL DESIGN

There are several key specifications for a PLL that play important roles in various applications, for example, the phase noise, phase jitter, spurs, loop bandwidth (BW) and frequency resolution. In a particular application, one or more of these factors may be more crucial than others, but all of them are mutually dependent. The designer must understand the system impact and be able to make the right design tradeoffs that best suit the application [2].

Phase noise is a measure of spectral purity of a signal. In communication systems, phase noise and spurs degrade the quality of the TV pictures, limit the precision of satellite positioning and affect the detection of a channel in a hostile environment. In clock generation, phase jitter degrades the signal-to-noise (SNR) of a downstream analog-to-digital convertor (ADC). Loop BW is of particular importance in a frequency hopped system, since it determines the dynamic behavior of the PLL. In order to obtain

good phase noise, a wide band may be necessary to suppress the noise from the VCO. This would put stringent requirements on the noise from the reference and from loop components like the CP and the LPF. BW scales with reference frequency for loop stabilization. The reference frequency sets the resolution for frequency synthesis, which is predetermined by channel spacing. A fractional-N PLL is used to synthesize frequencies with an arbitrary precision from a fixed reference clock. This removes the direct link between loop bandwidth and frequency resolution. However a wide band reduces the effective oversampling ratio for the delta-sigma modulator (ΔΣ), which is used to control the fractional bits in the divider, and makes the quantization noise a much bigger noise contributor. If this noise is removed through additional loop filter poles, then a fractional-N PLL does not provide any BW advantage over an integer-N PLL when a similar reference frequency is concerned. The tradeoff between loop bandwidth and phase noise represents a fundamental limitation in fractional-N PLLs.

1.3 OUR CONTRIBUTIONS

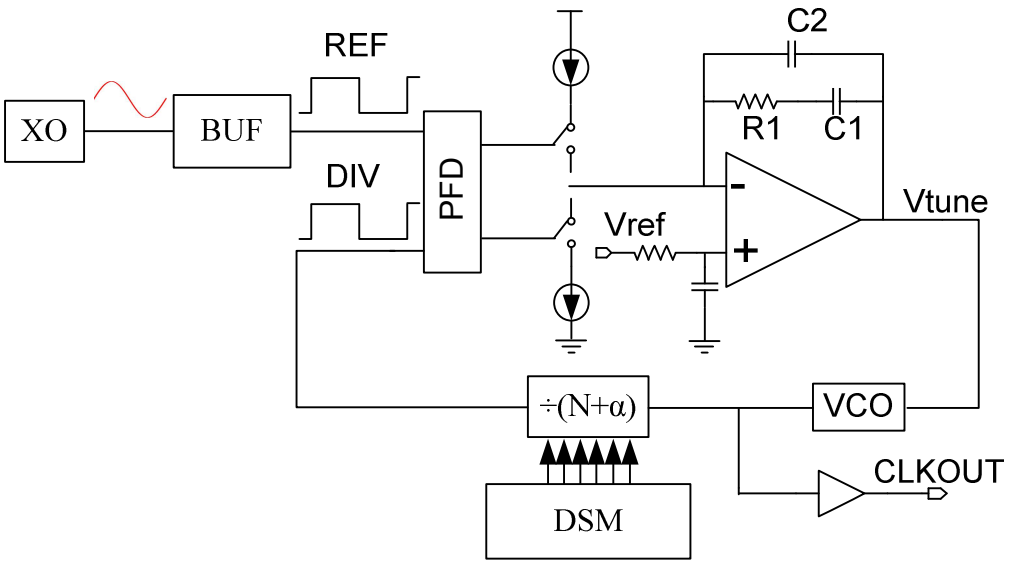
In this dissertation we present a novel PLL architecture using a simple clock edge multiplier to obtain a higher internal reference. The higher internal reference can be used for the benefit of bandwidth extension or phase noise reduction. Assuming a sinusoidal reference, the proposed method achieves 6x increase in update rate. A 6x increase in f_{update} means a 24dB reduction in quantization noise for the same loop BW. The phase noise reduction can be traded off for a 6x increase in loop BW if desired. In addition, we demonstrate how to lock a loop with an irregular reference, which has never been published before.

In this approach we use simple clock squaring buffers with built-in offset to extract multiple clock edges from one cycle of a sinusoidal reference. The magnitude of a

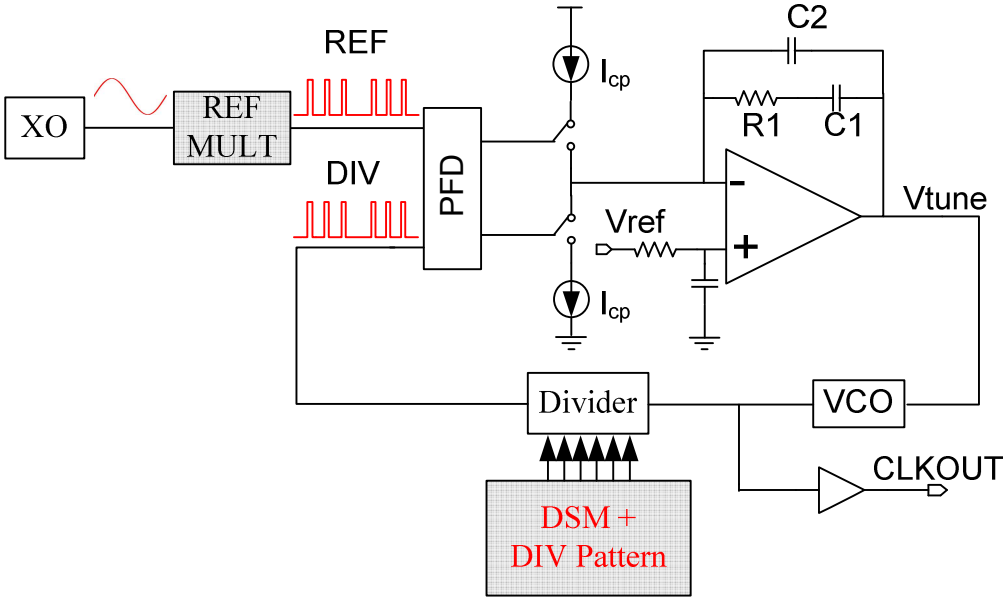
sine wave contains phase information with sufficient phase to voltage gain around the zero-crossings. The clock buffers are used like comparators. Multiple edges are derived from one period of the reference. Effectively this forms a reference frequency multiplier. Since the ΔΣ is clocked off this higher internal reference, this allows for higher oversampling ratio (OSR). A detailed description on the ΔΣ and OSR is given in Chapter 3. A higher OSR provides better quantization noise shaping and makes a wideband fractional-N PLL possible.

The new reference derived from a sine reference contains six edges per cycle (the number "6" comes from the symmetry of a sine wave, this will be discussed further in Chapter 4). The edges are clustered around the original zero-crossings and form an irregular period. This is due to the fact that phase to voltage gain peaks around the zero crossings. Further from the zero crossings, the gain gradually decreases till it flattens out, where no phase information can be extracted. The irregularity of the clock is taken into account in the divider. We program the divider to divide such that the loop achieves steady state when the divided clock matches the irregular reference.

In Figure 1.2 we compare our proposed fractional-N PLL with the conventional approach. The difference in architecture can be easily seen. Suppose f_{ref} is the reference frequency in the conventional approach, which equals f_{XO}, the frequency of the XO, then with our proposal we create an internal reference of $6 \cdot f_{ref}$. This frequency is the phase update rate at which the loop acquires and updates its error signal. This can be used to gain significant phase noise reduction, as we shall see in Chapters 2 and 3. The irregularity of the derived clock presents a challenge for the design. But because using a simple buffer to gain a 6x increase in the reference frequency brings a tremendous advantage in terms of quantization noise, we overcome the problem by imposing a fixed pattern to the divider. One issue associated with the fixed divider pattern is spurs. If the



(a) Conventional Fractional-N PLL



(b) Proposed Fractional-N PLL

Figure 1.2: Conventional Approach vs. Proposed Approach

fixed divider pattern does not match the reference pattern, there can be an increase in the spurious energy. For this we invented a calibration scheme that generates the perfect divider pattern to minimize spurs. The blocks that involve significant innovations are shaded with grey in Figure 1.2.

1.4 PRIOR WORK

1.4.1 Phase Noise Cancellation

A fractional-N PLL is typically implemented using a ΔΣ modulator to modulate the desired fractional number into a sequence of integers, so that over time the divided clock approximates the desired fractional ratio [2]-[12]. This creates significant noise on the feedback clock. This noise (due to quantization) must be sufficiently filtered and isolated from the VCO to achieve a good phase noise performance. A digital-to-analog converter (DAC) based phase noise and spur cancellation technique has been developed [4]-[11]. In this approach a DAC is used to convert the quantization error after 1^{st} order integration to current and then subtract it from the charge pump current. This technique requires a good dynamic range and linearity of the DAC. In [4] a 7 bit, segmented thermometric DAC was used in combination with a dynamic element matching technique to improve linearity. Gain matching between the DAC and CP also limited the quantization energy that remained after applying the technique. In [5] and [6], various calibration schemes are used to improve the technique. In addition, the matching requirement for a given performance increases drastically as the reference frequency is reduced. For example with reference frequencies of 35MHz, 48MHz and 50MHz, the approach achieves 15dB, 20dB and 29dB phase noise cancellation. A high reference frequency not only increases the power consumption, but it may also be unavailable in an

System-on-Chip (SoC) environment. The phase noise cancellation technique, though successful, is quite complicated in actual implementation.

1.4.2 Fractional-N PLL through multi-phase VCO

We have so far described multi-modulus fractional-N frequency synthesis where fractional division is achieved by using a ΔΣ modulator to modulate the desired fraction into a sequence of integers at an oversampling rate. This technique is capable of producing frequencies with arbitrary resolution, but the minimum phase step at the divider output is still equivalent to one VCO cycle. Another way of achieving fine frequency resolution is through the use of multiphase VCO [13]–[15], in which an analog interpolation is used to produce multiple phases. This allows the minimum phase jump at the divider to be Δ/n, where n is the total number of phases in the VCO. Since quantization noise power is proportional to Δ^2, theoretically a 4x phase interpolation results in 12dB phase noise reduction. Phase mismatches among the multiple VCO phases can limit the performance and in [13] a dynamic element matching technique was used.

1.4.3 Fractional-N All Digital PLL

All-digital PLL (ADPLL) frequency synthesis has emerged as a promising alternative to analog PLLs. In ADPLL a time-to-digital converter (TDC) is used to measure phase replacing the PFD and CP in traditional approach. A wideband PLL requires a linear TDC with fine quantization steps. TDC is typically implemented using an inverter chain or a Vernier delay line. The variations in the delay element produce fractional spurs, making a wideband ADPLL spurious performance still lagging that of an analog PLL. Calibration schemes have been developed to circumvent this problem [16]-[20].

1.5 SUMMARY

The tradeoff between loop bandwidth and phase noise represents a fundamental limitation in a fractional-N PLL. Our main contribution is to use a simple buffer for reference multiplication. This increases the OSR and allows for more efficient quantization noise shaping. The technique provides 24dB phase noise reduction or 6X improvement in loop BW. A calibration scheme is also invented to reduce spurs. The spurious energy arising from mismatches between the irregular cycles and divide pattern can be effectively calibrated out. The comparison between the proposed approach and the state of the art for quantization noise reduction is shown in Table 1.

Techniques	Phase noise reduction	Reference Frequency	Complexity	Need Calibration?	Limitations
DAC based phase noise cancellation [5]	25dB*	12MHz	High	Yes	High power
Multiphase VCO [13]	12dB**	40MHz	Medium	Yes	High power
ADPLL with Digital cancellation [18]	15dB*	50MHz	High	Yes	Need long calibration cycle
This work	24dB**	4MHz	Medium	Yes	Irregular Reference

Table 1: Comparison between our work and the state of the art for quantization noise reduction

* Measured result
** Simulated result

Chapter 2: PLL Fundamentals

2.1 Phase Domain Loop Analysis

The PLL is a nonlinear system which poses difficulties for the conventional analysis method of using a transfer function, especially during lock acquisition. However in the locked condition, a linear time-invariant (LTI) model can be used, assuming the PFD transfer characteristic is linear in this operating region [29]. A phase-domain LTI model is shown in Figure 2.1, where the PFD is modeled with a linear gain of $I_{CP}/2\pi$, and K_v is the VCO gain in [Hz/V]. The ideal integration $1/s$ accounts for frequency to phase conversion. Here we have assumed that the VCO response time is much faster than the loop bandwidth, as is typical, and no pole is modeled in the VCO gain. A second order passive loop filter formed by R_1, C_1 and C_2 is used as an example for ease of analysis. The loop gain can be found as:

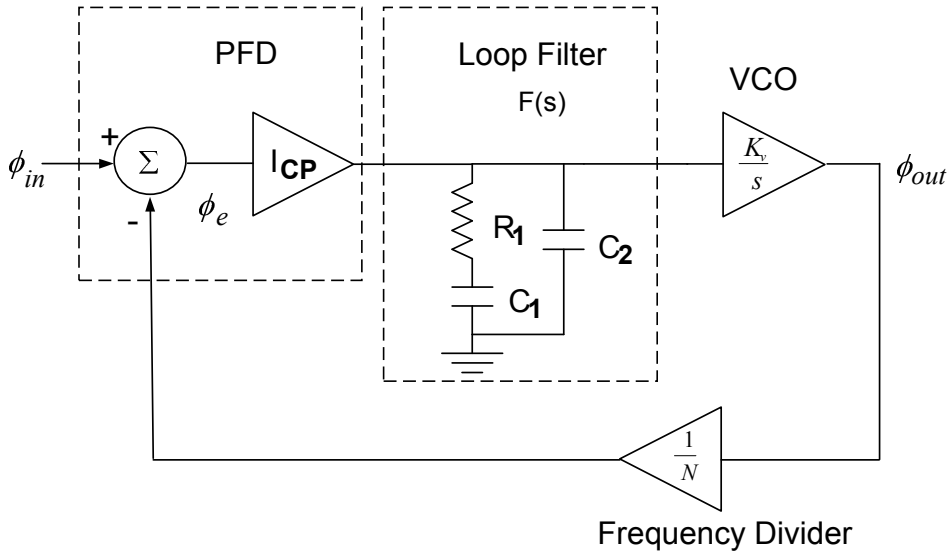


Figure 2.1: Phase Domain Linearized Model

$$G(s) \approx \frac{V_{DD}}{\alpha M R_1} \frac{K_v}{S^2 C_1} \frac{1+SR_2C_1}{1+SR_2C_2} \qquad \text{assuming } C_1 >> C_2 \qquad (2.1)$$

This is a third order, type II loop with two integrator poles at DC. We approach a third order loop problem with a second order approximation by neglecting the high frequency pole at *1/(R1C2)*; this allows for an exact closed-form solution. Note in Eq. (2.1) the 2π factor in the PFD gain is cancelled by the VCO gain given by $2\pi \cdot K_v$ in [rad/V]. Omitting the high frequency pole, the close loop transfer function becomes:

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{\frac{I_{CP}K_vR_1}{N}s + \frac{I_{CP}K_v}{NC_1}}{s^2 + \frac{I_{CP}K_vR_1}{N}s + \frac{I_{CP}K_v}{NC_1}} \qquad (2.2)$$

Rewriting this transfer function in the equivalent standard second order form we obtain:

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (2.3)$$

By definition we have:

$$\omega_n = \sqrt{\frac{I_{CP}K_v}{NC_1}} \qquad (2.4)$$

$$\zeta = \frac{1}{2}\sqrt{\frac{I_{CP}K_vC_1}{N}}\ R_1 \qquad (2.5)$$

ω_n is called the natural frequency of the system. It is the resonant frequency of the un-damped system. ζ is called the damping factor. A damped system prevents excessive peaking. Rewriting the loop gain in terms of the above defined parameters, we have:

$$G(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2} \tag{2.6}$$

The unity gain frequency is found by setting $|G(j\omega_u)| = 1$ and the closed loop -3dB frequency is found by setting $|H(j\omega_{3dB})| = \sqrt{2}$, respectively:

$$\omega_u = \sqrt{2\zeta^2 + \sqrt{4\zeta^4 + 1}}\,\omega_n \approx 2\zeta\omega_n \qquad \text{except for small } \zeta \tag{2.7}$$

$$\omega_{3dB} = \sqrt{2(1 + 2\zeta^2)}\,\omega_n \approx 1.5\omega_u \qquad \text{except for small } \zeta \tag{2.8}$$

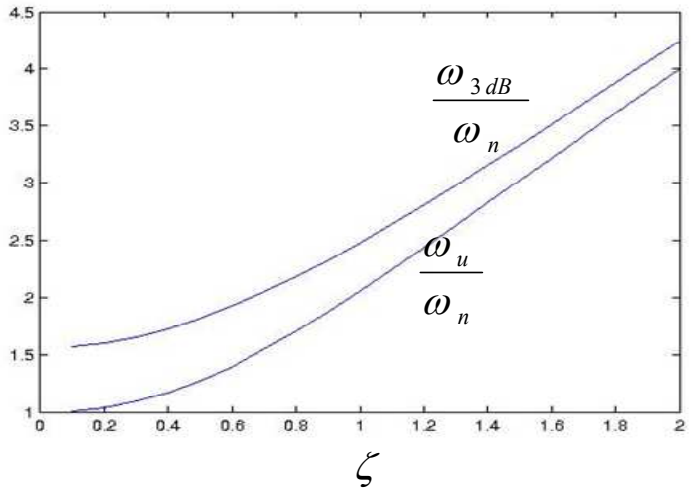


Figure 2.2: Unity gain BW and -3dB frequency vs. the damping factor

The last approximation is used in design to quickly estimate the closed loop bandwidth. It is based on the plot in Figure 2.2 [30]. Equations (2.4), (2.5), (2.7) and (2.8) are used in design to find the loop filter components. K_v is heavily technology dependent and is usually not a design parameter; it is determined through circuit simulation. The Bode plot for loop gain is shown in Figure 2.2. Due to the integrator in the loop filter,

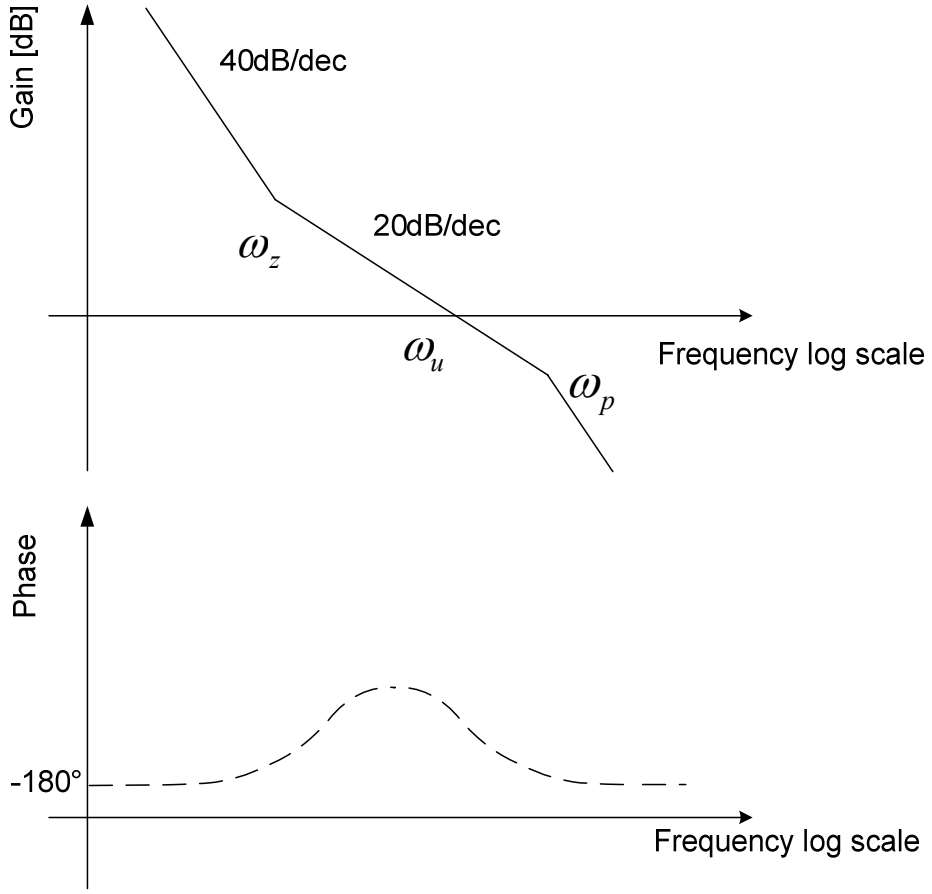


Figure 2.3: Bode Plot of Open Loop Gain

the loop gain initially has a -40 dB/dec slope, and the compensating zero brings it to -20 dB/dec where it crosses over zero. The high frequency pole is used for high frequency suppression and should be placed outside the unity gain bandwidth (UGBW) to maintain sufficient phase margin. For optimum phase margin and transient response, ω_u is set approximately at the geometric mean of the zero and the pole, $\omega_u = \sqrt{\omega_z \omega_p}$ [2]. Frequently the zero is placed at 1/3 of ω_u, and the pole is at 1/3 to 1/4 of ω_u. To

maintain loop stability ω_u needs to be between 1/10 and 1/5 of the reference frequency for integer-N PLLs, and typically much smaller for fractional-N PLLs. Based on these conditions, the loop filter components can be designed.

2.2 PHASE NOISE AND JITTER

Phase noise is a measure for spectral purity of a signal. In the time domain representation, it is the random fluctuation of the phase of a waveform due to timing "jitter". Jitter by definition is the variation of the significant instants of a clock signal from their ideal positions in time. This is also referred to as "total jitter "or "absolute jitter" [31]. As illustrated in Figure 2.4, we have:

$$\{j_n\} = T(n) - nT \tag{2.9}$$

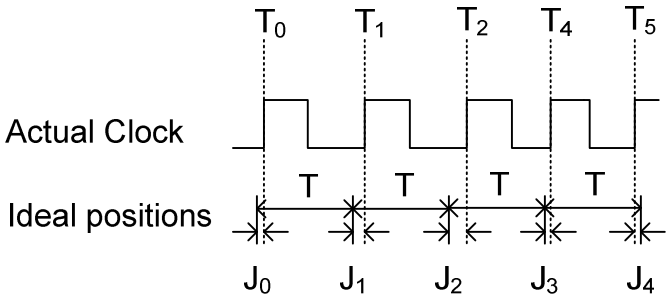


Figure 2.4: Clock Jitter

Absolute jitter can be derived from phase noise by integrating the total area under the phase noise curve. The lower integral limit is usually application specific. On the other hand, phase noise can also be derived by taking the power spectral density of jitter, $\{j_n\}$, assuming $\{j_n\}$ is a stationary stochastic process. Power spectral density is a measure of energy distribution over frequency. For a wide-sense stationary random

process, it is defined as the Fourier transform of the autocorrelation function. Cycle-to-cycle jitter and period jitter are in general the integrals of far-out phase noise. In applications like communication systems phase noise is more of a concern; while in digital clock generation, for example, timing jitter is more important. Phase noise is measured as the noise power in a unit bandwidth at an offset frequency ω_m away from the carrier ω_0, divided by the carrier power, resulting in the Single-Side-Band (SSB) noise density in [dBc/Hz]. It is related to power spectral density (PSD) on the phase, $S_\theta(\omega)$, in [rad^2/Hz] by:

$$L(\omega_m) = \frac{S_\theta(\omega_m)}{2} \tag{2.10}$$

This can be understood as follows. An ideal oscillator produces a perfect sine wave corresponding to a Dirac impulse at ω_0, with phase fluctuation $\theta(t)$, it becomes:

$$V_{out}(t) = A \cdot \sin(\omega_0 t + \theta(t)) \tag{2.11}$$

This phase fluctuation causes the power to spill into adjacent frequency bins and forms a skirt on the voltage spectrum (see Figure 2.5). Consider a sinusoidal tone in the phase, $\theta(t) = \theta_m \cdot \sin(\omega_m t)$, assuming small angle modulation, $\theta_m << 1$, we have [2], [3]:

$$V_{out}(t) \approx A\sin\omega_0 t + \frac{A}{2}\cos\omega_0 t \cdot \theta_m \sin\omega_m t \tag{2.12}$$

Applying some trigonometric identities it becomes:

$$V_{out}(t) \approx A\sin\omega_0 t + A\frac{\theta_m}{2}\left[\sin(\omega_0 + \omega_m)t + \sin(\omega_0 - \omega_m)t\right] \tag{2.13}$$

From Eq. (2.13), it can be seen the output spectrum of the oscillator consists of narrow FM side bands at $\omega_0 \pm \omega_m$. We can relate the PSD for the oscillator output Vout to the Phase noise PSD:

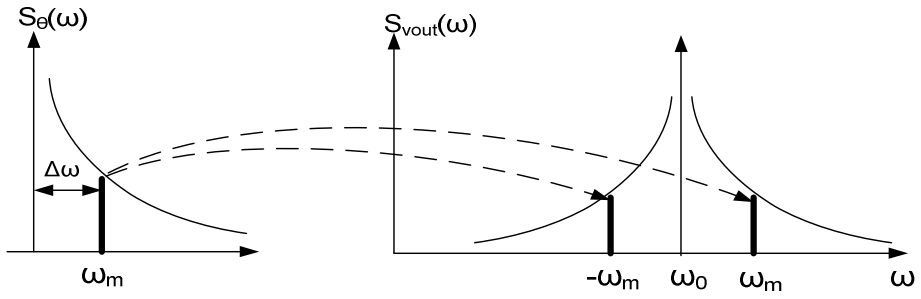


Figure 2.5: Phase Noise

$$S_\theta(\omega) = \frac{\theta_m^2}{2}\delta(\omega - \omega_m) \tag{2.14}$$

$$S_{V_{out}}(\omega) = \frac{A^2}{2}[\delta(\omega - \omega_0) + \frac{S_\theta(\omega - \omega_0)}{2} + \frac{S_\theta(\omega_0 - \omega)}{2}]$$

$$= \frac{A^2}{2}\delta(\omega - \omega_0) + \frac{\theta_m^2}{8}A^2\delta(\omega_0 + \omega_m) + \frac{\theta_m^2}{8}A^2\delta(\omega_0 - \omega_m) \tag{2.15}$$

This analysis can be generalized to the entire phase noise skirt as long as $\theta_m << 1$. For larger θ_m, Equation (2.11) can be expanded using Bessel functions [3]. In this manner the total phase noise skirt is translated to side lobes at both sides of the carrier frequency. And the phase noise at ω_m offset frequency is calculated relative to the carrier, in log scale as:

$$L(\omega_m) = 10\log\left(\frac{S_{V_{out}}(\omega_0 + \omega_m)}{A^2/2}\right) = 10\log(\frac{S_\theta(\omega_m)}{2}) = 20\log(\frac{\theta_m}{2}) \tag{2.16}$$

Since frequency is the derivative of phase we find the PSD for frequency deviation from the phase noise as:

$$S_{\Delta f}(\omega) = \omega^2 \cdot S_\theta(\omega) = 2\omega^2 \cdot 10^{L(\omega)/10} \tag{2.17}$$

We are now set to derive the integrated phase jitter calculated from phase noise. Suppose the band of interest is between f_1 and f_2, the RMS phase error integrated for this band is then given (in rad^2) as:

$$\Delta\Phi^2{}_{rms} = \int_{f_1}^{f_2} S_\theta(2\pi f_m)df_m = \int_{f_1}^{f_2} 2 \cdot 10^{L(2\pi f_m)/20} df_m \tag{2.18}$$

The RMS jitter (in seconds) is then:

$$T_{rms,noise} = \frac{\Delta\Phi_{rms}}{2\pi f_0} = \frac{\sqrt{\int_{f_1}^{f_2} 10^{L(2\pi f_m)/20} df_m}}{\sqrt{2}\pi f_0} \tag{2.19}$$

This also applies to spurs. Suppose a spur located at f_{spur} is at L_{spur} (in dBc), the jitter due to this spur is calculated as [31]:

$$T_{rms,spur} = \frac{10^{L_{spur}/20}}{\pi f_0} \tag{2.20}$$

Total jitter is then:

$$T_{rms,total} = \sqrt{T^2_{rms,noise} + T^2_{rms,spur}} \tag{2.21}$$

2.3 How Phase Noise and Spurs Affect a System

In modern communication systems PLL synthesizers are frequently used as the

local oscillator (LO) signal on the receive side to down convert an RF signal band to IF. The phase noise and spurious performance of the LO signal are particularly important in the receiving path. Excessive phase noise results in reciprocal mixing which limits the receiver sensitivity. In the process of down conversion, the phase noise of the LO will transfer on to the IF signal, and this is true even if the signal in the RF band is noiseless. This can become worse if a large blocker is present in the vicinity of a small RF signal of interest. The reciprocal noise due to the blocker would dominate the noise at IF. This is illustrated in Figure 2.6. In addition, the spurs on LO can directly cause channel corruption by mixing two adjacent channels down to the same IF frequency. This happens if a spur tone is located off LO by the same offset Δf as an adjacent channel is displaced in the RF band. This is depicted in Figure 2.7 [32].

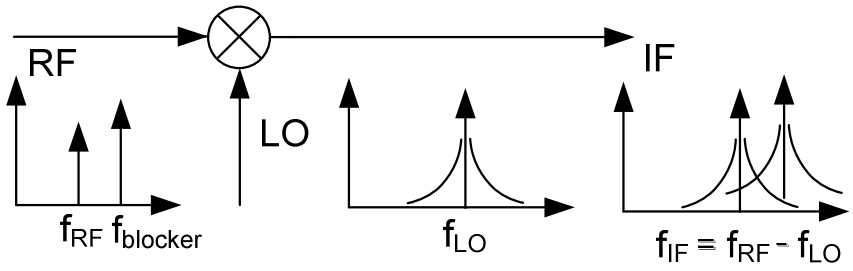


Figure 2.6: Reciprocal Mixing

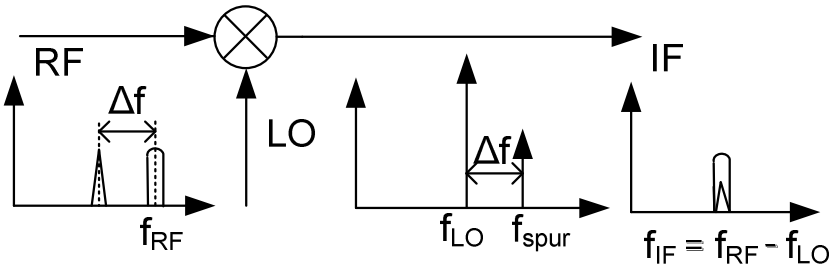


Figure 2.7: Channel Corruption due to LO Spur Tone

On the other hand, when a PLL is used for clock generations in an SoC, the integrated phase jitter is more of a concern. For example, a PLL is generating a high speed sampling clock for an ADC as shown in Figure 2.8. Clock jitter causes a displacement of the precise sampling moment away from the ideal clock instance, and leads to a sampling error, which is proportional to the magnitude of the jitter and the slope of the continuous time signal. The SNR due to sampling jitter can be found as [33]:

$$SNR_{jitter} = 20\log(2\pi \cdot f_{in} \cdot T_j / f_s) \qquad (2.22)$$

where f_{in} is the input signal frequency, f_s is the sampling frequency and T_j is the RMS jitter.

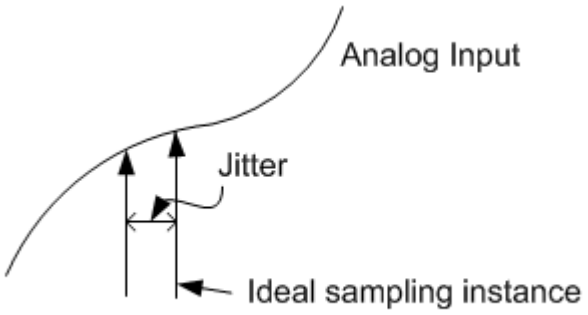


Figure 2.8: PLL Phase Noise Leads to Sampling Clock Jitter

2.4 NOISE SOURCES IN A PLL

As we have seen phase noise plays an important role in systems where PLLs are used. The most important noise sources in a PLL are the VCO noise and the reference noise. In addition, PFD, CP and the divider each contributes noise to the final clock. Fundamentally these are due to either thermal noise or 1/f noise. The effects of supply

and substrate activity due to digital switching circuits are often treated as noise as well. In this subsection, we use the LTI model derived in Section 2.1 to calculate the noise transfer function from each noise source to the PLL output [2], [3], [29]. The results provide important guidelines in process of making design tradeoffs.

The LTI model from Figure 2.1 is re-plotted here in Figure 2.9 with an explicit noise source added for the reference and the VCO. The VCO noise is modeled as the additive noise at the output. The transfer function from $\phi_{n,vco}$ to ϕ_{out} is found to be:

$$\frac{\phi_{out}(s)}{\phi_{n,vco}(s)} = \frac{1}{1+G(s)} = \frac{sN}{sN + K_{PD} \cdot F(s) \cdot K_v} \tag{2.23}$$

Here the PD gain and loop filter gain are represented by K_{PD} and $F(s)$ respectively. Similarly the transfer function from $\phi_{n,ref}$ to ϕ_{out} is:

$$\frac{\phi_{out}(s)}{\phi_{n,ref}(s)} = \frac{G(s) \cdot N}{1+G(s)} = \frac{N \cdot K_{PD} \cdot F(s) \cdot K_v}{sN + K_{PD} \cdot F(s) \cdot K_v} \tag{2.24}$$

We can see the noise from VCO is high-pass filtered with pass band gain of 1 while the noise from reference is low-pass filtered with gain of N. For both the transition band edge is the PLL loop BW. The multiplication factor of N comes from the fact that output frequency is N times larger than the reference.

For a free running VCO, the zero-crossing times follow a random walk process and the jitter referenced to a fixed starting point grows unbound with the measuring time interval. There is a squared root dependence given in Eq. (2.25).

$$j(nT_s) \approx K_v \sqrt{\frac{n}{2}} \cdot \frac{T_s^{3/2}}{2\pi} \sigma_n \tag{2.25}$$

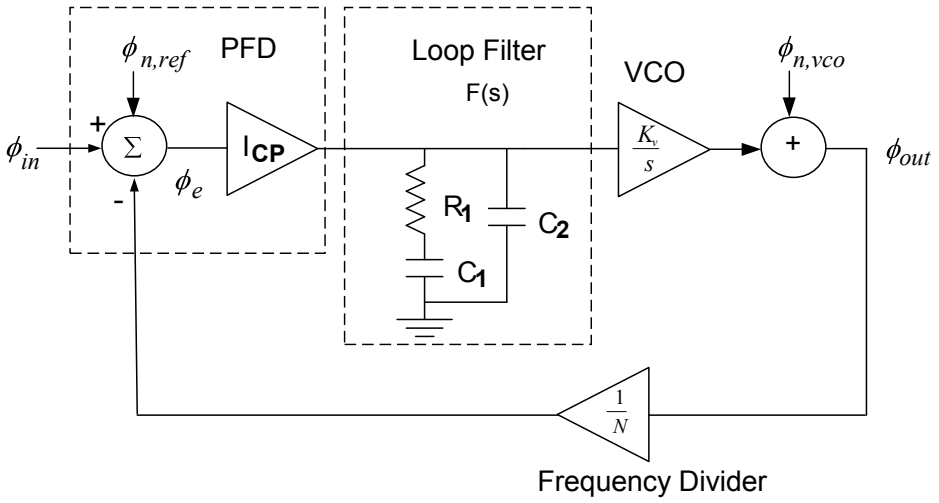


Figure 2.9: Noise Model for PLL

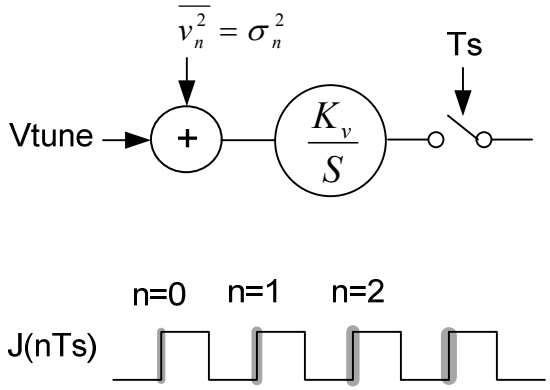


Figure 2.10: Free Running VCO Noise

This is described in Figure 2.10, where "v_n" represents the instantaneous device noise (both thermal noise and 1/f noise in nature) with a variance of σ_n^2, referred to the input of VCO [34], [35]. Jitter is measured at the output. Once the VCO is placed inside a PLL, as the time interval grows much larger than the reciprocal of the PLL loop BW, the loop tracks out the cumulative jitter from VCO and the RMS jitter becomes bounded. This is shown in Figure 2.11, where ω_L is the closed loop PLL BW. Similarly the phase

noise of the VCO and the XO reference in the cases of open and closed loop are shown in Figure 2.12 [21], [22], [23], where $f_L = \omega_L / 2\pi$.

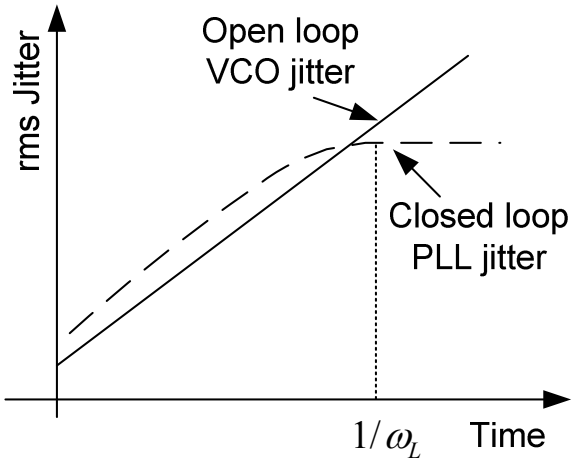


Figure 2.11: Open and Closed Loop Jitter

The noise of PD, CP, the LPF and the divider can also be analyzed by referring to either the reference node or the VCO node, and by use of the corresponding noise transfer function. The CP usually contributes little noise due to the small duty cycle factor in steady state. In steady state the UP and DN pulses from the PD are nearly equal and only last for a brief moment to switch the CP on. In our design, the minimum pulse width for UP and DN signals is about 200 ps. Only during this brief moment can the internal noise from the CP make it to the PLL output. Overall the effect is averaged out by the reference period [2]. However the switching activities of the PD could cause supply ripples that are responsible for producing reference spurs. An LDO can be used to stabilize the supply and improve the spurious performance [59]. The noise from the LPF and OPAMP also contribute to overall noise through various low-pass and band-pass functions and requires careful design tradeoffs.

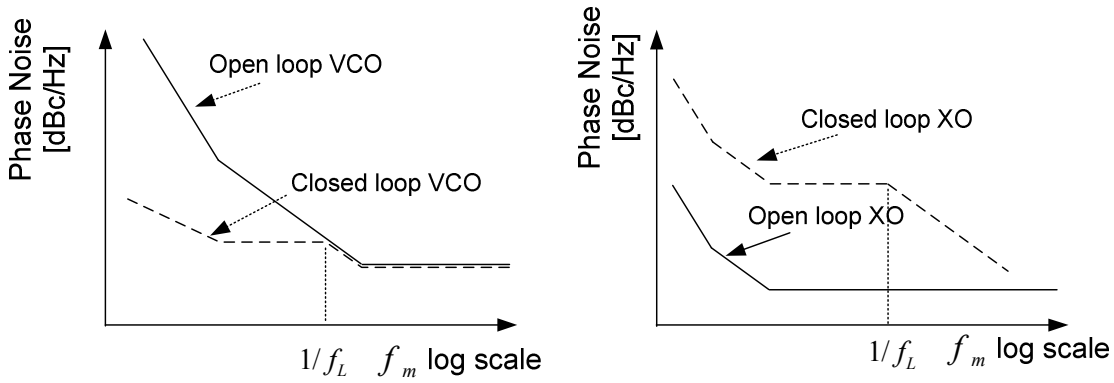


Figure 2.12: Open and Closed Loop Phase Noise

2.5 SUMMARY

In this Chapter a phase domain LTI model for a PLL is introduced. The LTI model is often used in practical PLL design and in analyzing phase noise and jitter. Phase noise is the frequency domain representation of random fluctuations in the phase angle; jitter is the time domain representation of essentially the same phenomena. Various jitter components can be derived by integrating phase noise over a particular frequency band. In communication applications, phase noise limits the precision of a receive channel through reciprocal mixing. Distinctive spur tones can cause direct channel corruption. In clock generation, phase noise and spurs degrade the SNR in a subsequent ADC. Various noise sources are identified and their impacts on the overall PLL performance are carefully analyzed.

Chapter 3: Quantization Noise in Fractional-N PLL

3.1 Introduction

Fractional-N PLLs are a well-known technique to synthesize fine resolution frequency sources from a fixed reference signal. Arbitrarily spaced frequency sources can be synthesized without the need to use a low frequency reference as is the case in integer-N PLLs. This removes the direct tradeoff between loop bandwidth and frequency resolution. Since loop bandwidth (BW) generally scales with the reference frequency, this opens up possibilities of building a wider band loop. In a popular approach, a ΔΣ modulator is used to modulate the desired fractional number into a sequence of integers so that over time the divided clock approximates the desired fractional ratio [2]-[11]. As a byproduct of this process, and for fractional-N PLLs in general, the instantaneous clock edges are never truly locked, except perhaps only momentarily. Nevertheless, the loop achieves steady state by aligning the time average of the divide-down clock to the reference clock. This creates significant noise on the feedback clock. This noise (due to quantization) must be sufficiently filtered and isolated from the VCO to achieve a good phase noise performance. If this noise is removed through additional loop filter poles, then fractional-N PLL does not provide bandwidth advantage over integer-N PLL with similar reference frequency. This defeats the purpose of fractional-N frequency synthesis. The tradeoff between loop bandwidth and phase noise represents a fundamental limitation in fractional-N PLLs. Over the years, several techniques have been successfully developed for BW enhancement. Next we briefly review these techniques.

3.2 Quantization Noise Basics

A conventional fractional-N PLL is shown in Figure 3.1. A ΔΣ modulator generates a sequence of integers *y[n]*, which modifies the divider's instantaneous divide

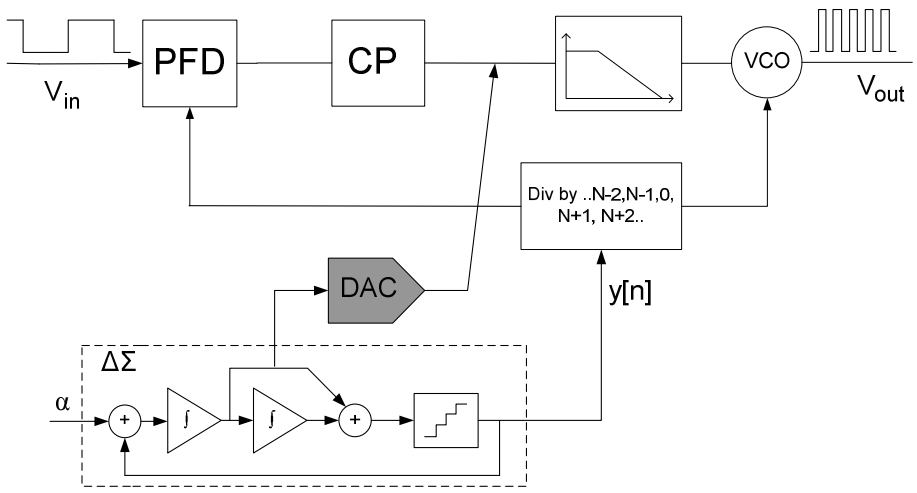


Figure 3.1: A Conventional Fractional-N PLL

ratio to $N+y[n]$. The divided down clock is compared to a clean reference f_{ref}. A frequency sensitive phase detector (PFD) first drives the frequency difference to zero, phase error is then measured using a charge pump by taking the timing difference between the corresponding zero-crossings of f_{ref} and f_{div}. This error signal is used to drive the remainder of the loop and gets updated once every reference cycle. Under locked condition $f_{vco} = (N + \alpha) \cdot f_{ref}$, where α is the desired fractional ratio. The ΔΣ modulator serves to predict α using a sequence of integer numbers $y[n]$, and shape the quantization noise energy toward high frequencies without adding a fixed pattern to it, such that $e_Q[n] = y[n] + \alpha$. Here $e_Q[n]$ is the zero-mean, high-passed quantization noise. It is the raw quantization error resulting from a quantizer inside the ΔΣ modulator. In general the raw quantization errors are treated as white noise, uncorrelated with the input. This allows a linear time-invariant model for quantization noise analysis using noise transfer function (NTF) [24]. The quantization noise adds to the overall phase noise at the output; the effect can be accurately modeled as shown in Figure 3.2 [36].

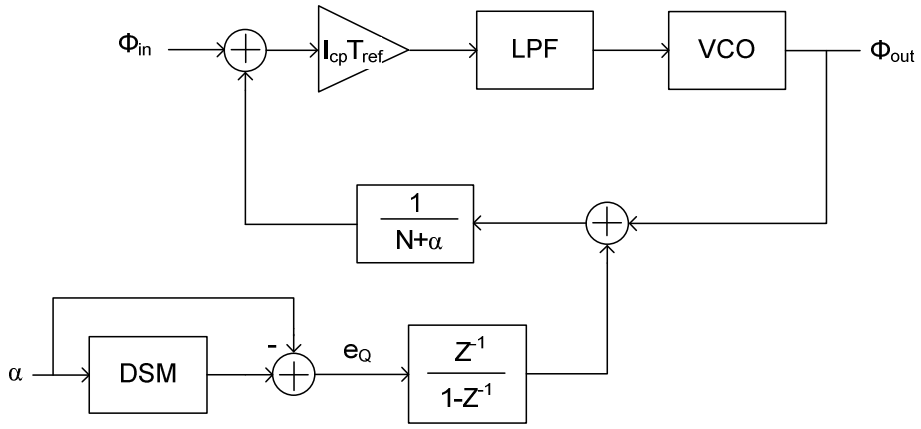


Figure 3.2: Linearized Model for Quantization Noise

Under white noise assumption, the quantization noise power is $\Delta^2/12$, Δ is the minimum quantization step. In a sampled system with sample rate of f_{ref} the entire noise power folds in the band of 0 to f_{ref}, and the power spectral density becomes $\Delta^2/(12 \cdot f_{ref})$. The phase noise contribution of the $\Delta\Sigma$ at the input of the divider is then given as:

$$S(z) = \frac{\Delta^2}{12 \cdot f_{ref}} \left| NTF(z) \right|^2 \frac{(2\pi)^2}{\left| 1 - z^{-1} \right|^2} \tag{3.1}$$

When referred to the PLL output, it becomes:

$$S_{\Phi_{out}}(z) = \frac{(2\pi)^2}{12 \cdot f_{ref}} \left| 1 - z^{-1} \right|^{2(n-1)} \cdot \left| \frac{G(z)}{N + \alpha} \right|^2 \tag{3.2}$$

$G(z)$ is the PLL closed loop transfer function, $N+\alpha$ is the nominal frequency ratio. Here the NTF of an nth order $\Delta\Sigma$ modulator is given by $(1-z^{-1})^n$ for ease of analysis and without loss of generality. This corresponds to an n-fold zero placed at DC in the NTF. The quantization step Δ is taken to be 1. Notice in the above equations the noise shaping has lost one order due to the rectangle integration in the z domain. This is to account for the conversion from frequency to phase [2]. While the $\Delta\Sigma$ modulator controls frequency,

here we seek to derive the power spectral density for phase. After replacing z with $e^{j2\pi f/f_{ref}}$ we have:

$$S_{\Phi_{out}}(f) = \frac{(2\pi)^2}{12 \cdot f_{ref}} \cdot (2\sin(\frac{\pi f}{f_{ref}}))^{2(n-1)} \cdot \left| \frac{G(z)}{N+\alpha} \right|^2 \tag{3.3}$$

The total integrated ΔΣ noise power that falls inside the PLL bandwidth becomes:

$$A_{n,\Delta\Sigma}{}^2 = \int_{-f_0}^{f_0} S_\Phi(f)df \approx \frac{1}{12}\frac{(2\pi)^{2n}}{2n-1}\left(\frac{2f_0}{f_{ref}}\right)^{2n-1} \tag{3.4}$$

where f_0 is the PLL bandwidth, and the approximation is taken for $f_0 << f_{ref}$. The factor $f_{ref}/2f_0$ is the equivalent OSR as in the conventional ΔΣ theory. For 1st order noise shaping, every doubling of the OSR reduces in-band noise by 9dB. In the proposed technique we increase the OSR by 6x and eq. (3.4) predicts a 24dB reduction for in-band noise. This can easily be seen in the simulation plots shown in Figure 3.3, where the quantization noise is filtered by the same PLL transfer function, modeled as a 3rd order, type II PLL. In one case the sample rate is 60MHz, and in the other 10MHz. The higher sample rate offers close to 24dB of noise reduction, as predicted in eq. (3.4). The choice of these numbers will become clear after we describe the reference multiplying scheme.

3.3 Quantization Noise Cancellation

Quantization noise arising from the ΔΣ modulator limits the BW in a fractional-N PLL. Quantization error is often approximated as white noise since its spectrum appears random for busy input signals, but strictly speaking, quantization error is a deterministic process and correlates with the input. In digital implementations of the ΔΣ modulator, quantization error is known a priori and can be used to cancel the error charges injected

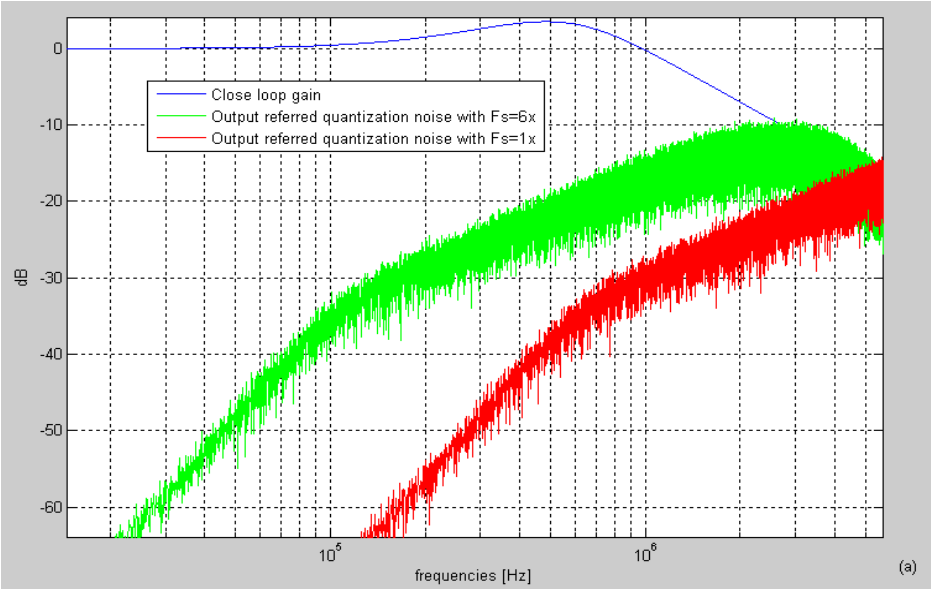


Figure 3.3: Quantization noise of a 2nd Order ΔΣ in a Type II PLL

into the loop filter, and no additional hardware is needed. The added cancellation path effectively isolates the VCO from the granularity of the divider, so the output can maintain the desired steady frequency with little phase modulation. This is known as the quantization noise cancellation approach in BW enhancement. Excellent references are found in [3]-[11]. A DAC is needed here to convert the pre-calculated quantization error to analog current, which is then subtracted at the loop filter summing junction. This is the DAC cancellation approach shown as the shaded path in Figure 3.4. DAC cancellation works by shielding the VCO control line from seeing the abrupt changes coming from the ΔΣ's attempt to approximate the desired fractional ratio. This approach is sometimes called analog interpolation [2].

Since the cancellation has to take place in the analog domain, the dynamic range, linearity and gain matching requirements of the DAC are crucial for complete or near complete cancellation. Noise shaping and dynamic element matching techniques are used

to improve the DAC resolution and linearity [3], [5]. Gain matching is improved using calibration schemes based on the sign-error LMS algorithm [4], [5].

Quantization noise degrades PLL performance and it must be attenuated before attempting to increase the loop BW. The quantization noise cancellation approach has proved to be an efficient method for BW enhancement but requires a DAC with good linearity and high dynamic range. Next we turn to another factor affecting the BW, the update rate, f_{update}.

3.4 PHASE UPDATE RATE

One effective way of mitigating the quantization noise is to increase the reference frequency (while keeping the loop bandwidth the same). This increases the oversampling ratio of the ΔΣ modulator, thereby pushing more of the quantization noise outside the PLL bandwidth. Unfortunately, increasing the input reference frequency is not an option in most applications since a fixed XO is used for reference.

To achieve an update rate higher than f_{ref}, a nonlinear block is necessary for frequency multiplication. To avoid significant phase noise degradation, this nonlinear block needs to be simple with a minimum number of devices. The DLL technique (delay locked loop) has been used for frequency multiplication to produce low phase noise clocks. In DLL, a reference clock is fed into a chain of uniform delay cells, the output from the delay chain is compared with the reference, and the resulting phase difference is used to adjust the delays till the two edges are exactly 2π apart. The delay cells form evenly spaced edges that span one reference cycle; these edges are combined into one faster clock [25]. Alternatively, a VCO is used as a re-circulating delay element, and the ring is periodically opened to align with the reference [26], [27]. However delay cells and edge combing logic can become the dominant noise source. In particular a DLL

working on an input reference of a few MHz (as is very common in practice) will require several stages with relatively long delays which would either need a large number of digital inverters or an analog delay block. Both of these options are area intensive and also add a significant amount of phase noise. In our proposed architecture, much simpler buffers are implemented to extract additional edges from a sinusoidal reference. We assume inexpensive, readily available crystal oscillators in the tens of MHz are used as reference.

3.5 A PLL WITHOUT CP AND DIVIDER

In an early study we considered a charge pump free, divider less fraction-N PLL [55], [56]. In this phase locking architecture, we assume that the phase information is not just available from zero-crossings. Instead it is extracted from reference magnitude at all times. This allows phase information to update at a rate much faster than that of the reference signal, and potentially a faster PLL can be built. For the purpose of illustrating the idea, imagine a saw tooth wave as the reference with frequency f_{ref}, with amplitude range from 0 to 2π volt. It describes an ideal situation of voltage representing phase. The reference signal is then converted through an ADC clocked by the feedback clock. The ADC produces a digital representation of the saw-tooth phase. Simultaneously, we can calculate on a cycle by cycle basis of what the phase should be under the locked condition. Starting from some initial offset, it advances by $(f_{ref} / f_{out}) \cdot 2\pi$ [rad] per VCO clock, where f_{out} is the output frequency, $f_{out} = (N + \alpha) \cdot f_{ref}$. The difference between sampled phase and predicted phase is the error signal that adjusts the frequency control on the VCO. Once locked, the feedback loop reduces this error signal to zero. The phase prediction block is nothing but a phase accumulator also clocked by the feedback clock. The word being accumulated each cycle represents $2\pi / N + \alpha$ radians. The phase predictor

essentially predicts where the sampled phase would have to line up under locked condition.

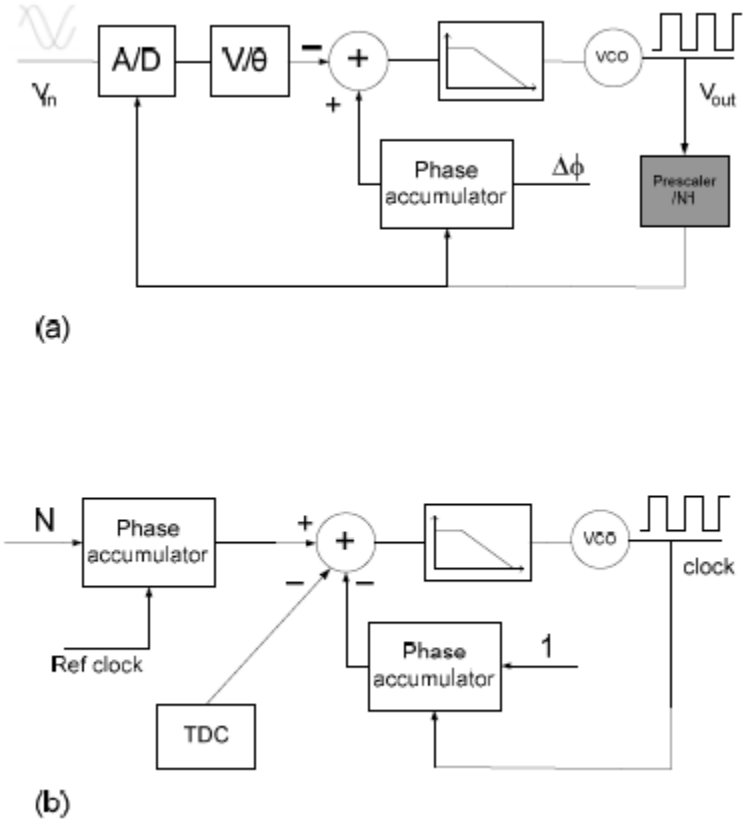


Figure 3.4: FNPLL architectures (a) Our Early work (b) an ADPLL

Typically a saw-tooth reference is not attainable for the desired frequency range. A sine wave could serve the purpose as well. But since a sine wave traverses the same amplitude twice per 2π, a cosine reference would be needed to resolve this ambiguity. Sampling both I and Q references can also avoid the flat part of the voltage vs. time transfer, and this alleviates the noise degradation problem once the voltage-to-phase conversion has taken place. For the moment, we shall assume that both the sine and cosine references are available, and that the ADC alternately samples one and the other.

The nonlinear voltage-to-phase conversion is done through a table-lookup ROM. The complete architecture is shown in Figure 3.4(a). The VCO is assumed to be digitally controlled, similar to the one given in references [28].

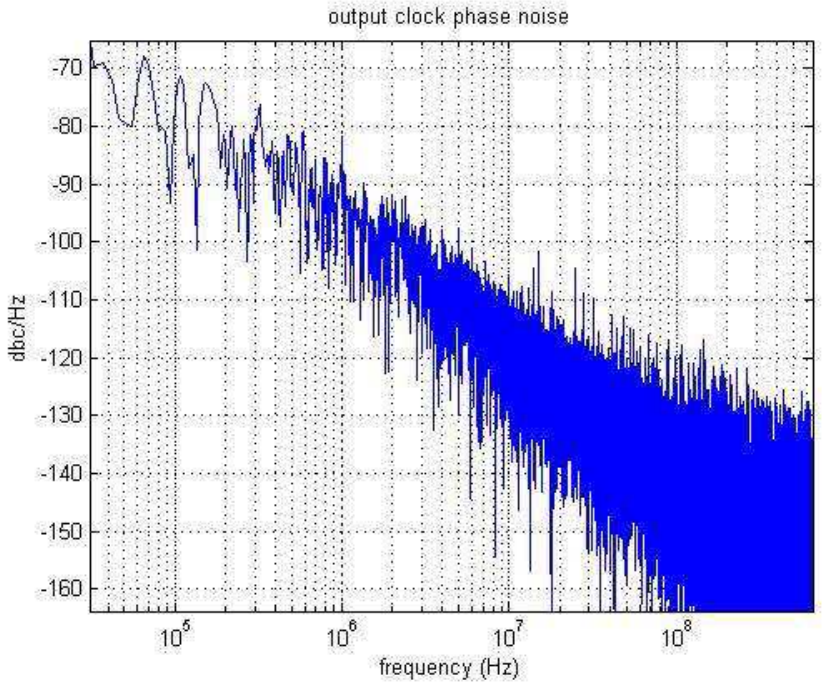


Figure 3.5: Simulated phase noise for the CP-free fractional-N PLL

In this PLL architecture, the ADC dynamic range becomes the bottleneck. The voltage resolution of the ADC directly translates into quantization in time. These quantization errors mix with the desired phase error, and the feedback loop ends up responding to both. To ensure that the quantization noise does not ruin the phase noise performance of the output clock, the ADC is required to have good dynamic range. Since the PLL output clock is targeting the GHz range operation, a high dynamic range ADC at this sampling range is difficult. An optimized design involves a simple prescaler to bring the ADC operating speed down. This is shown as the shaded block of Figure 3.4 (a). The

prescaler divides the VCO frequency by a fixed N1. The loop operates on the same principle as discussed, and the only difference is that the phase being accumulated is now $N1$ times larger. Figure 3.5 shows the phase noise simulation for f_{out} =1.22GHz, f_{ref} =10MHz, $N1$=16, $f_{update} = f_{out} / N1$. A 10-bit flash ADC running at f_{update} (about 75MHz) is used.

The phase domain transfer function is derived using the small signal model shown in Figure 3.6 [55]. t(k) represents the output clock transition timestamps, k=1,2,3.., is the clock transition index number; the input is modeled as a continuous phase, 2πfref·t, sampled by the feedback clock t(k). After the analog-to-digital conversion,

$$\theta_{in} = 2\pi f_{ref} \cdot t(k) + e_{ADC} + \theta_0 \tag{3.5}$$

Here e_{ADC} is the quantization noise of the ADC, θ_0 is the initial phase offset when t(k)=0. Similarly,

$$\theta_{out} = 2\pi f_0 \cdot t(k) \tag{3.6}$$

Phase accumulator acts like an integrator to the phase step, but when the clock transition t(k) is considered the input, the transfer function is simply a ramp, given by:

$$\frac{\theta_{ref}(k)}{t(k)} = k\frac{2\pi}{N} = k\frac{2\pi f_{ref}}{f_0} = 2\pi f_{ref} \cdot kT_0 \tag{3.7}$$

T_0 is the nominal output clock period. Eq. (3.7) indicates that the accumulator block correctly predicts the edges where the sampled phase would have to line up under locked condition. This is also the reason its output is designated as θ_{ref}. In a sense, the feedback clock operates on the input sinusoids to generate θ_{in}, and θ_{in} acts like the feedback phase. Setting up the closed loop equations in the z-domain, we have:

$$(\theta_{ref} - \theta_{in}) \cdot \frac{K_L}{1 - z^{-1}} = \theta_{out} \tag{3.8}$$

VCO is modeled as a digital integrator; K_L is the loop gain in [Hz/rad]. Replacing θ_{in} with (3.5) and dropping the arbitrary initial phase to simplify the equations, we obtain the transfer function as:

$$\frac{\theta_{out}(z)}{\theta_{ref}(z)} = \frac{\theta_{out}(z)}{e_{ADC}(z)} = \frac{K_L G(z)}{(1 + K_L G(z)/N) + z^{-1}} \tag{3.9}$$

G(z) represents a digital loop filter, K_L is the loop gain in [Hz/rad]. We can see that the transfer function is a low-pass function with a DC gain of N, as expected.

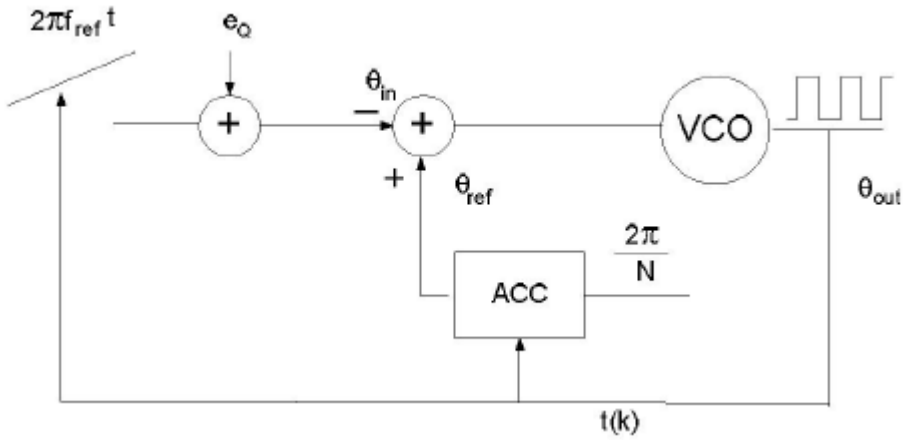


Figure 3.6 Small Signal Model for Our Early Work

An interesting parallel can be observed between our CP-free PLL and an all digital PLL described in [28]. The ADPLL is symbolically shown in Figure 3.4(b). The VCO phase is measured by accumulating a "1" on every output clock edge. Once locked, the expected phase of the VCO can be calculated by accumulating the desired fractional ratio $N_{frac} = N + \alpha$ on each reference clock. The difference between the calculated phase and measured phase forms the error signal, and is determined using a time-to-digital

converter (TDC). The TDC is built as an array of inverter-delay elements and flip-flops. The inverter-delay based PD introduces a dead-zone the size of one inverter delay. For 130 nm CMOS technology this delay is typically 30 ps, but varies as much as 100% with process and temperature. As a result, this approach suffers a slow update rate (which is the same as f_{ref}) and with an output clock jitter window equal to the dead-zone. By comparison, in our PLL architecture, a digital word representing $2\pi/N_{frac}$ is accumulated; the instantaneous phase of the VCO is measured against the sine reference using an ADC, both events happen at the rate of f_{update}, which is much faster than f_{ref}. Consequently a faster PLL can be built. Due to the ADC dynamic range bottleneck issue discussed above we have decided to pursue a new architecture for this dissertation. This new architecture is presented in Chapter 4.

3.6 SUMMARY

Fractional-N PLL provides a convenient way to synthesize frequency with arbitrary resolution from a fixed reference. This is typically done using a ΔΣ to modulate the desired fractional ratio into a sequence of integers at an over sampled rate. The quantization noise presents a limitation for fractional-N synthesis. Phase noise cancellation technique has been developed to overcome the problem, but a successful implementation of this approach requires a good matching between the cancellation path and signal path. On the other hand a higher update rate would provide a more efficient quantization noise shaping through higher OSR. The high-passed quantization noise can be filtered out by the loop. This serves as the motivation for our work.

Chapter 4: Proposed Fractional-N PLL

4.1 General Idea

A precision reference for a PLL is typically generated using a quartz crystal resonator, which produces a sine wave. We utilize the fact that phase information is not only available at the zero-crossings of the sine wave, but also embedded in the voltage slope near the zero-crossings. Similar to a saw tooth reference, we have a case of voltage representing phase. If we use comparators with properly managed thresholds, multiple clock edges can be extracted from the sine wave. The general idea of the proposed approach is illustrated in Figure 4.1. Suppose we have five comparators each with a threshold voltage relative to the XO magnitude as marked on the sine wave. The comparators produce a set of duty-cycle skewed clocks shown below the sine wave. Running through a one-shot logic element the comparator outputs can be combined into one clock which on average is 10x the original XO frequency.

Contrary to a saw tooth reference a sine wave flattens out near 90° and 270°. Since the phase to voltage gain at these locations is close to zero, no phase information can be extracted. As a result, the new clock takes on a bursty look. We propose to lock the loop using this faster but irregular clock. This is achieved by programming the divider with a divide pattern which anticipates the irregularity of the reference. The purpose is to produce a feedback clock which matches the incoming pattern locally. While the loop gets updated at each time instant according to the irregular clock events, we must maintain noise shaping for fractional errors for good phase noise performance. The details are described in Section 4.3. The divide pattern repeats with each cycle of the original XO signal. A set of cyclic shift registers can be used to produce the pattern.

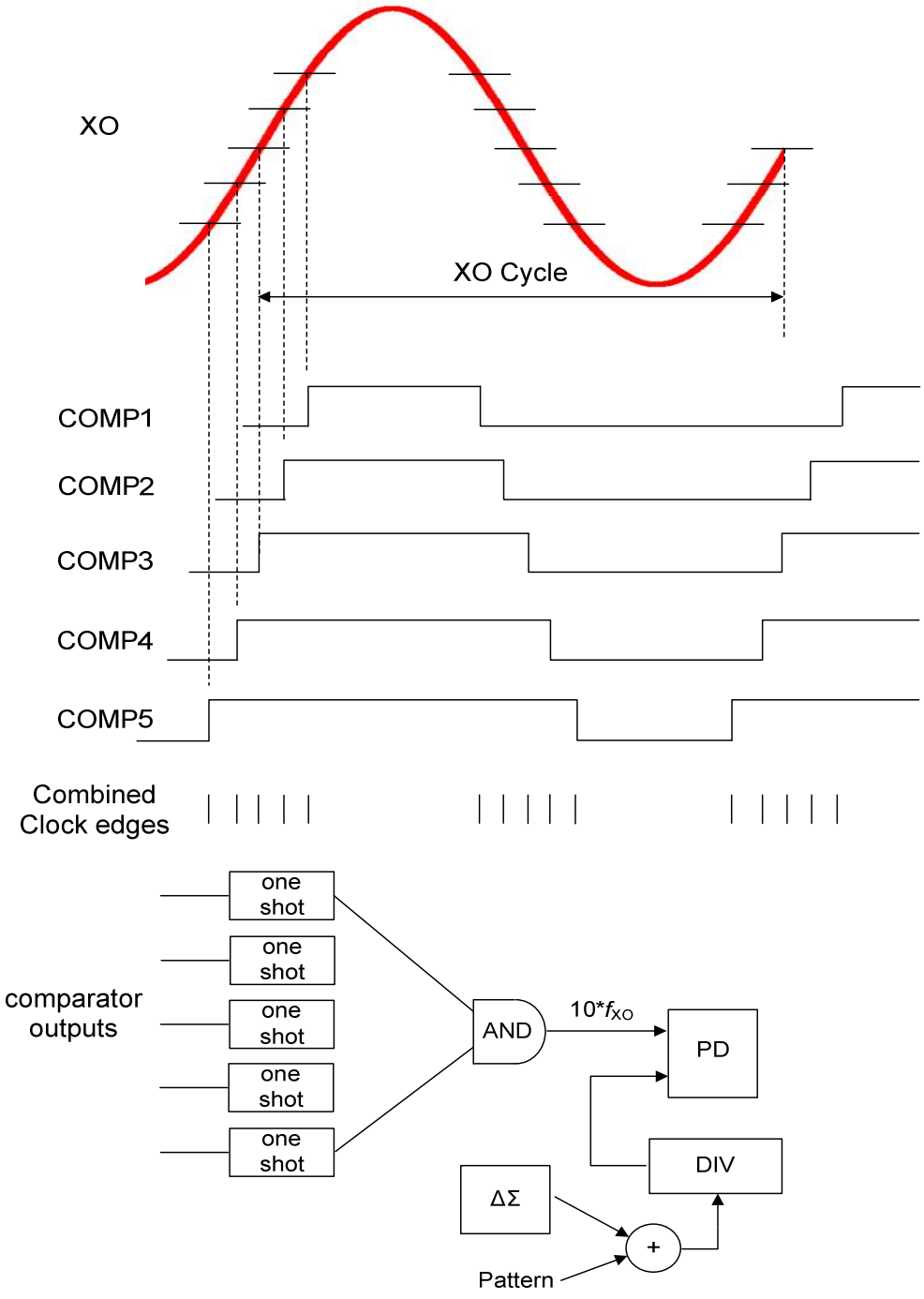


Figure 4.1: The General Description for the Proposed Approach

4.2 REFERENCE MULTIPLIER

Quantization noise degrades PLL performance and it must be attenuated before attempting to increase the loop BW. The quantization noise cancellation approach as described in the earlier Chapter has proved to be an efficient method. In this work we developed a simple technique to extract six edges from one period of a sinusoidal input reference, and update the phase detector 6x faster than the reference frequency, effectively forming a reference multiplier. This allows for better and more efficient noise shaping and moves most of the quantization noise beyond the PLL bandwidth. For a second order delta sigma modulator, a 6x increase in OSR theoretically results in 24dB reduction in quantization noise [57], [58]. While many nonlinear blocks perform frequency multiplication, noise degradation is the key factor to watch for. Since a well-designed PLL will have an in-band noise performance following that of the reference signal, the nonlinear block intended to achieve higher phase update rate must also not degrade the noise performance significantly from the original reference.

An on-chip reference is typically generated by placing a high quality crystal in the feedback path of an amplifier, generating a sinusoidal signal which is processed by a squaring buffer to produce a clock. In this work, we propose to use additional buffers with built-in offsets to extract multiple edges from the amplitude of a sine wave. The schematic is shown in Figure 4.2. A balanced buffer with equal driving strength of the PMOS and NMOS devices puts the threshold at the input zero crossing. In this approach, we simply take advantage of the magnitude information near the zero-crossings and produce more edges for phase update using similar squaring buffers, but with embedded offsets. As long as we keep the thresholds near the zero-crossings, noise degradation is negligible. Periodic steady state noise analysis was performed to simulate the integrated

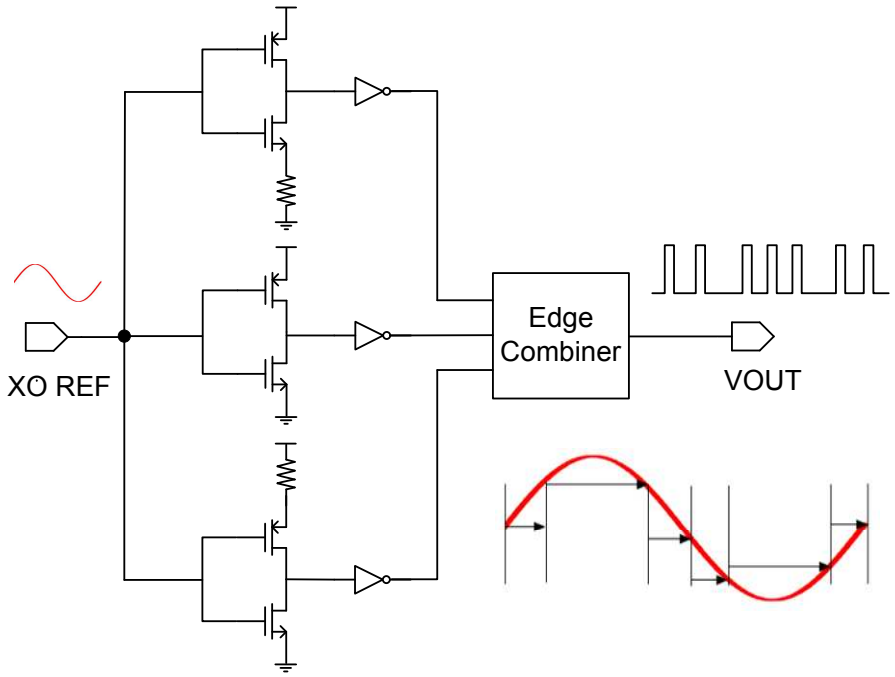


Figure 4.2: Reference Clock Edge "Multiplier"

noise at each transition point and referred to time domain to obtain the RMS jitter. Figure 4.3 shows the results.

We now obtain a new clock which has 6 edges per cycle, but the edges are clustered and the new clock assumes an irregular period. This becomes the new reference with which we attempt to lock the loop. The issue we must address first is whether noise shaping can be preserved when an irregular clock is applied to a ΔΣ. Since there is no published record on this issue, we took a simulation approach and the simulation results are detailed in Section 4.4. We conclude that the irregular clock does not degrade noise performance significantly compared to when an ideal clock with equal average frequency is used, given that the variations among these cycles are not so much as to overload the modulator. Intuitively this can be understood as follows.

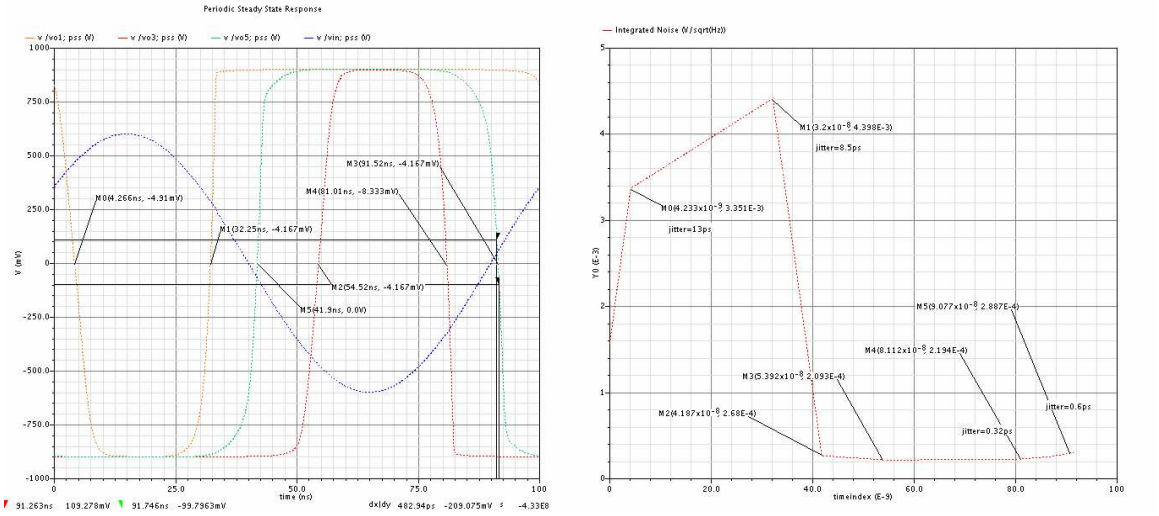


Figure 4.3: PSS simulation to calculate RMS jitter on the clock edge "multiplier"

We start with a second order ΔΣ modulator [4] as shown in Figure 4.4 (a), and the irregular clock shown in 4.4 (b). In a digital implementation, the same clock is used to update the state and the output of the modulator, and both are saved in registers. There is no distinction between a longer bit cycle and a shorter one since each enters the loop exactly once at the active clock edge. This largely preserves the noise shaping property. In addition, since the loop gets updated more frequently during T_0, T_2, T_3 and T_5 (thus higher OSR), and less so during T_1 and T_4 (thus lower OSR), the overall effect approaches that of an ideal clock. This is the fundamental limitation of this approach. The problem can be overcome by simultaneously employing a sine and a cosine as reference. In many systems both I and Q signals are readily available.

4.3 DIVIDE PATTERN

The bursty nature of the clock must be taken into account in the divider to ensure robust lock. We need program the divider to divide by a smaller number when there are more edges and divide by a larger number when there are fewer. To see how to incorporate a divide pattern in the output of the ΔΣ, let us take a look at the new reference

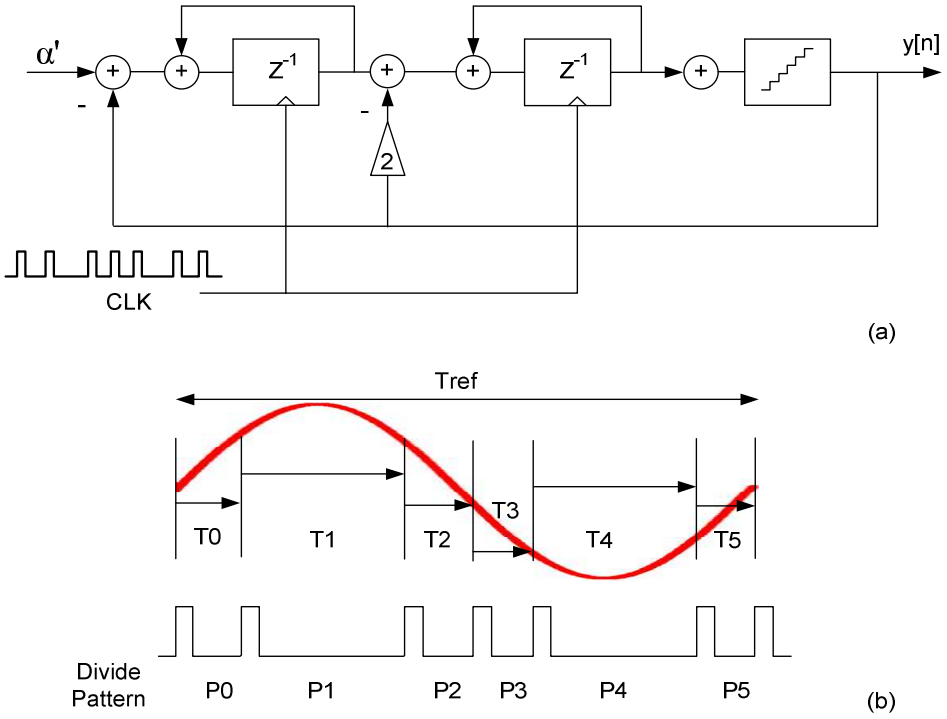


Figure 4.4: (a) Delta-Sigma Modulator and (b) Divide pattern

shown in Figure 4.4 (b). There are six irregular periods from T_0 to T_5 spanning one reference cycle, $T_{ref} = T_0 + T_1 + T_2 + T_3 + T_4 + T_5$. Assuming the desired synthesized frequency $f_{vco}=(N+\alpha)\cdot f_{ref}$, by replacing f_{ref} with $1/T_{ref}$ we have:

$$N+\alpha = \frac{T_0}{T_{vco}} + \frac{T_1}{T_{vco}} + \frac{T_2}{T_{vco}} + \frac{T_3}{T_{vco}} + \frac{T_4}{T_{vco}} + \frac{T_5}{T_{vco}} \qquad (4.1)$$

The first period T_0 contains P_0 VCO cycles, and second contains P_1 VCO cycles and so on. In total there are $N+\alpha = P_0 + P_1 + P_2 + P_3 + P_4 + P_5$ VCO cycles, $P_i = T_i/T_{vco}$ for $i=0$ to 5. The $\{P_i\}$ form an ideal cyclic divide pattern. Since each P_i is in general a fractional multiple of the VCO period, a quantized version is needed to control the divider. The details are described in Figure 4.5. We start by defining a new ratio $N'+\alpha' = f_{vco}/f_{avg}$, where f_{avg} is the average frequency of the irregular clock, for the reference multiplier

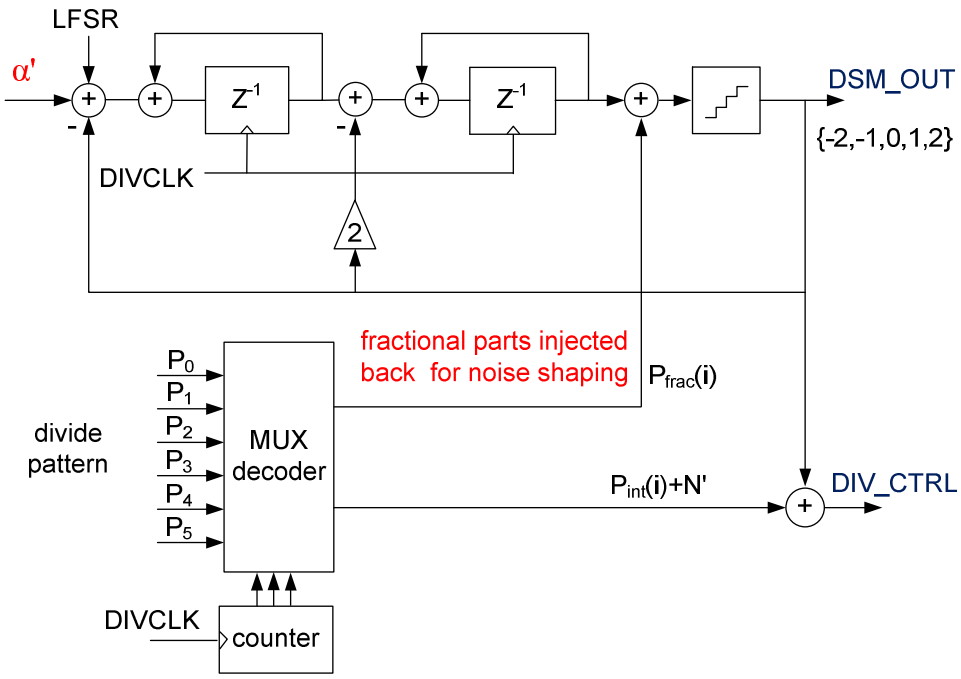


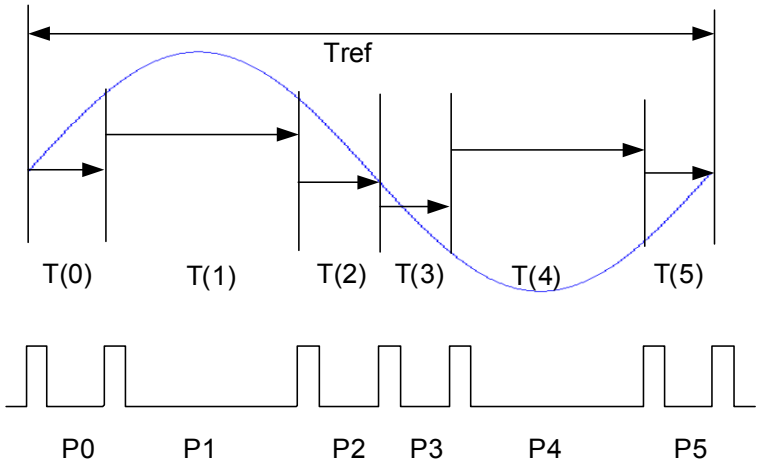
Figure 4.5: Generating the divide pattern

described in Section 4.2, $f_{avg}=6 \cdot f_{ref}$. It follows that $N'+\alpha'= (N+\alpha)/6$, N'=floor$((N+\alpha)/6)$ and $\alpha'=(N+\alpha)/6-N'$. Note α' is the new fractional ratio and input to the modulator with signal transfer function of 1. Next we derive a zero-mean pattern given by $P_i - (N'+\alpha')$ for $i=0$ to 5, and separate the sequence into integer parts $P_{int}(i)$ and fraction parts $P_{frac}(i)$. The integer parts are summed with N' outside the ΔΣ loop to form a divide pattern that allows the feedback clock to locally match the incoming pattern. The fractional parts are injected back to the loop before the quantizer to gain noise shaping. It is necessary to maintain zero DC on the fractional pattern in order to set the PLL frequency correctly. An example on how to generate the zero-mean sequences is shown in Figure 4.6. In the example $T(i)$ is the irregular clock period of the reference; $P(i)$ is the ideal divide pattern. The zero-mean pattern is obtained and separated into $P_{int}(i)$ and $P_{frac}(i)$. Not there is more than one way to derive $P_{int}(i)$ and $P_{frac}(i)$, and the calibration loop described in Section 4.4 will

eventually determine the sequence which minimizes the spurious energy. The proposed PLL is shown Figure 4.7.

Example

f_{vco}: 100.123xf_{ref}
f_{ref}: 10MHz
f_{avg}: 60MHz



		zero-mean pattern		
T(i) [ns]	P(i)	P(i)-(N'+α')	P_{int} (i)	P_{frac} (i)
12.5005	12.5159	-4.1713	-4	-0.1713
24.999	25.0297	8.3425	8	0.3458
12.5005	12.5159	-4.1713	-4	-0.1713
12.5005	12.5158	-4.1714	-4	-0.1713
24.999	25.0297	8.3425	8	0.3458
12.5005	12.5159	-4.1713	-4	-0.1713

Figure 4.6: an Example

4.4 System Level Simulations

Matlab and Simulink tools were initially used to establish the viability of this architecture. The PLL model in Simulink is shown in Figure 4.8 [37]. A10 MHz sine

wave was used and six clock edges are derived from angles near 0°, ±45°, ±135° and ±180° based on the sine wave amplitude information. This is done using the reference

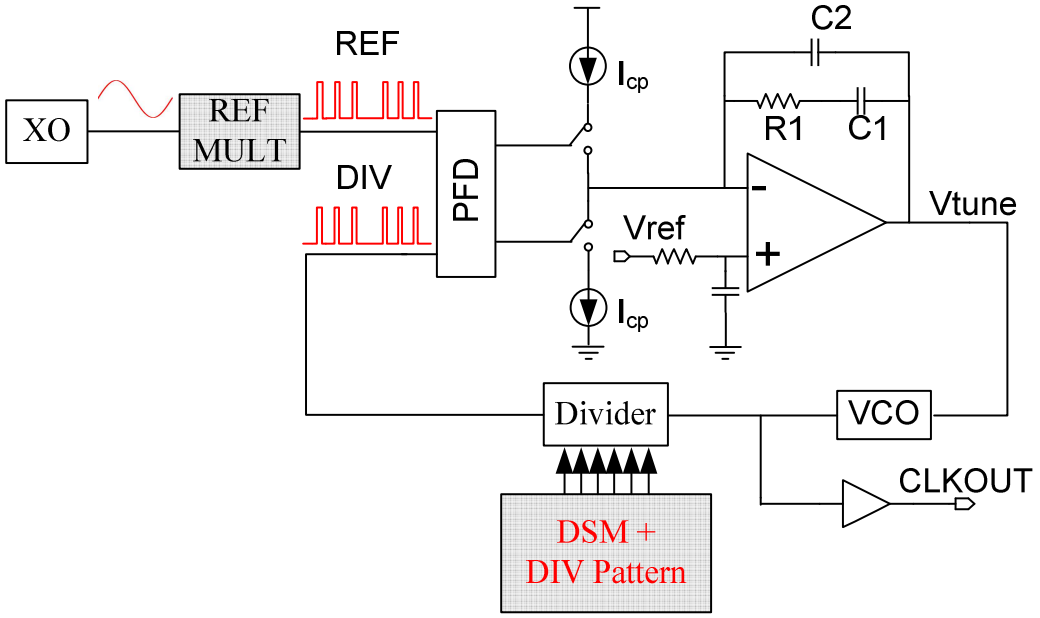


Figure 4.7: Overall Proposed PLL

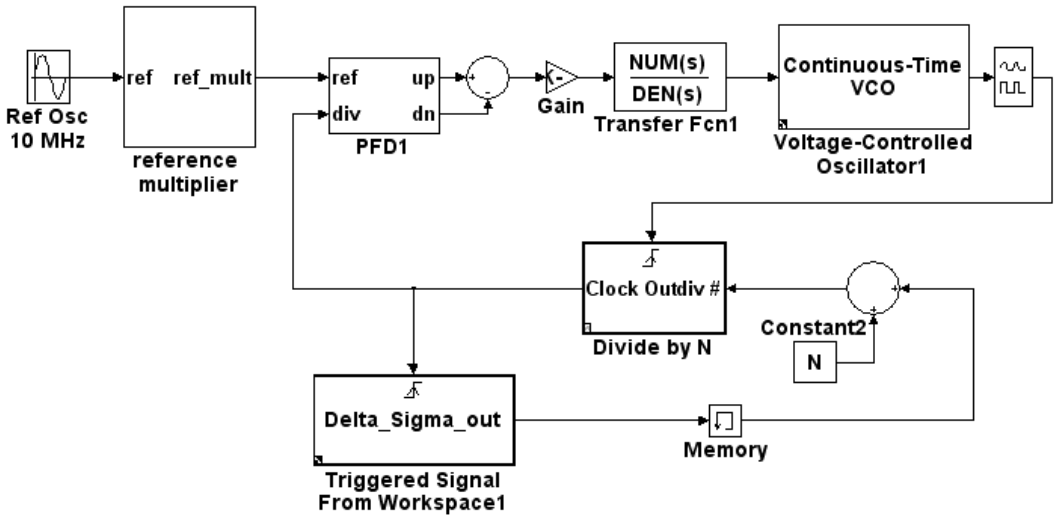


Figure 4.8: Simulink Model for Proposed PLL

multiplier which was modeled using comparators as shown in Figure 4.10. Random offsets were also added to the trip point of each comparator, and this is discussed in the Section 4.5 on calibration. The PFD was modeled using two Flip-Flops as it is commonly implemented. A continuous VCO model from the Communication Blockset was used in the simulation. The divider model is shown in Figure 4.10. The ΔΣ was modeled in a separated Matlab file and its output sequence was read in using a triggered signal port. The separate modeling approach allows for a faster simulation.

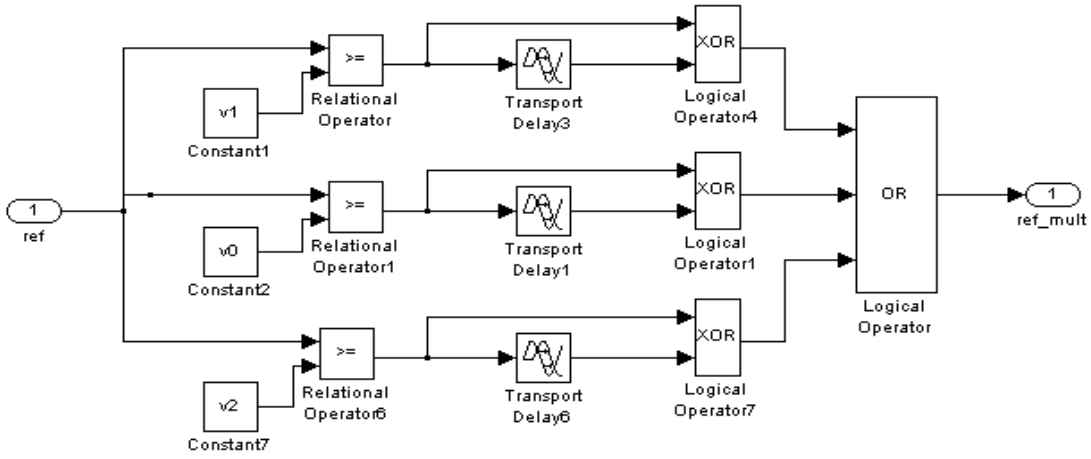


Figure 4.9: Simulink Model for Reference Multiplier

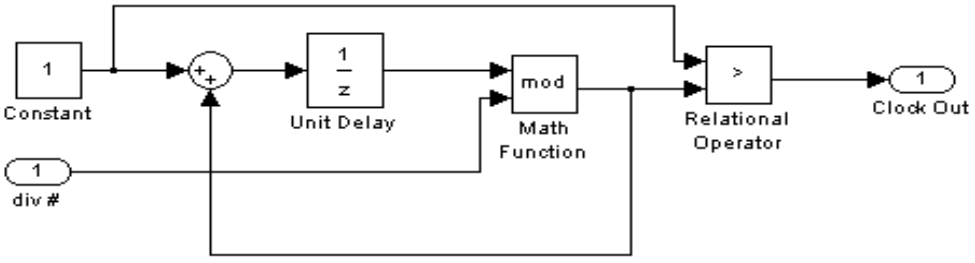


Figure 4.10: Simulink Model for Divider

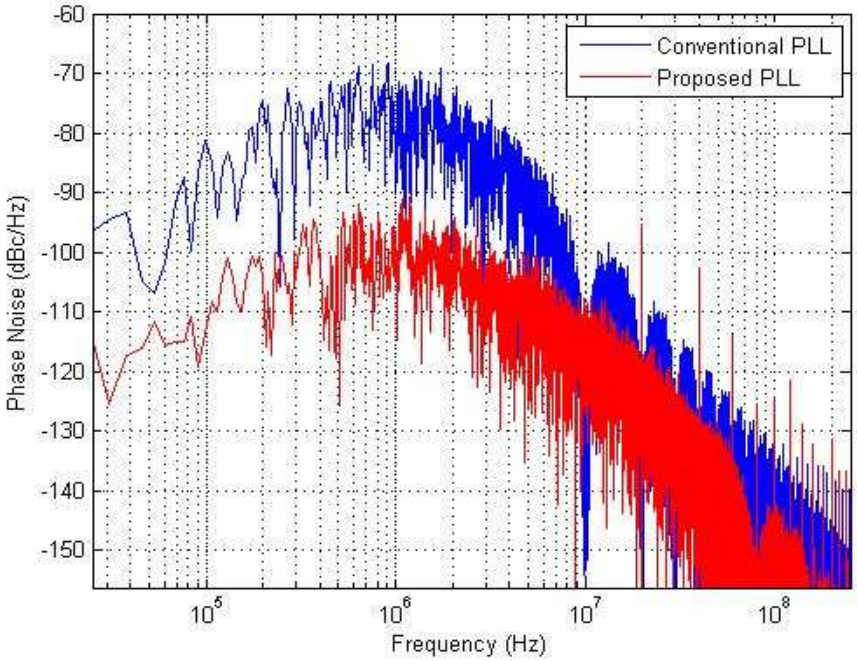


Figure 4.11: Phase noise of proposed system compared with conventional PLL

Conventional system with ideal Reference						
f_{ref}	f_{vco}	I_{cp}	K_{vco}	R1	C1	C2
10 MHz	$100.123 \cdot f_{ref}$	10 uA	2.5 GHz/V	10 KΩ	300 pF	10 pF
Proposed system with irregular reference						
f_{avg}	f_{vco}	I_{cp}	K_{vco}	R1	C1	C2
60 MHz	$16.6872 \cdot f_{avg}$	2 uA	2.5 GHz/V	10 KΩ	300 pF	10 pF

Table 2: Loop filter components

The simulations were carried out for two cases, one for the proposed PLL, and the other for conventional PLLs using an ideal 10 MHz reference. Loop BW was kept equal for ease of phase noise comparison. Figure 4.11 shows the phase noise plot calculated from these simulations, and loop filter components listed in Table 2. The absolute jitter of the output clock is measured and collected for 2^{17} samples under locked condition. From this time domain jitter the SSB power spectrum density is then calculated and normalized

to 1 Hz bin. We can see that the proposed system has over 20dB in-band phase noise reduction compared with a 10 MHz conventional PLL, approaching the theoretical limit predicted in (3.4). It shows that the irregular reference system does not degrade noise performance significantly when compared to the ideal 60 MHz system. However additional spurs are introduced. The lowest spur occurs at the original reference frequency of 10 MHz (a symmetry in the reference multiplier model pushes the spur to 20 MHz in Figure 4.11). Note this noise reduction can be traded off for bandwidth. Figure 4.12 shows the transient response on Vtune for the proposed PLL. In contrast it also shows transient response on Vtune for a conventional PLL with a 10 MHz reference. In the locked state, the divided clock matches the irregular reference clock up to a quantization error. The locked waveforms are displayed in Figure 4.13.

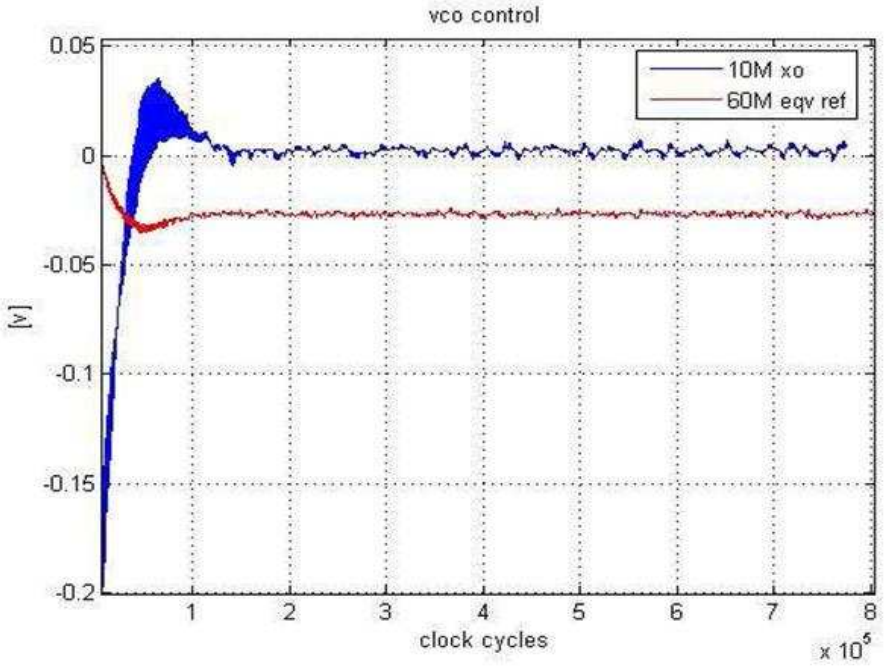


Figure 4.12: Vtune Transient Response

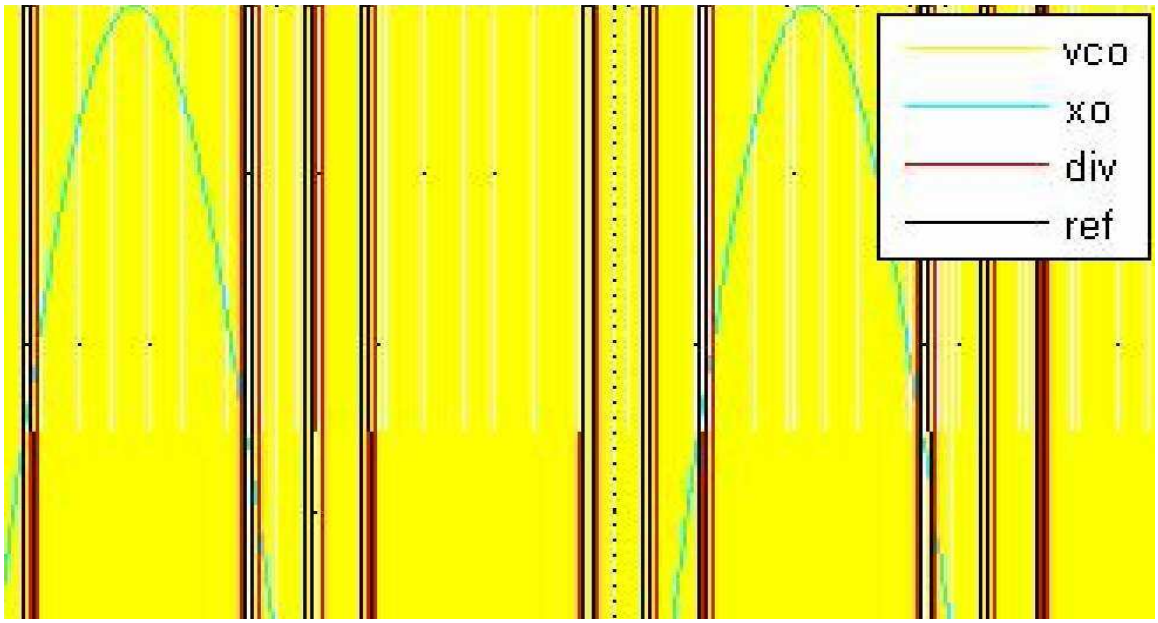


Figure 4.13: Locked Waveforms

4.5 SUMMARY

In this chapter we presented the new PLL architecture which forms the main body of this dissertation. We use simple clock buffers with built-in offsets to extract six edges per period of a sinusoidal reference. The edges are combined into a faster, but irregular clock to drive the rest of the PLL, effectively forming a reference multiplier. On average this clock is 6x faster than the original reference. The irregularity of derived clock is taken into account in the divider control. This is done by adding a fixed divide pattern along with the ΔΣ control bits. The integer pattern allows the feedback clock to locally match the incoming waveform, while the fractional pattern is injected back to the ΔΣ loop to gain noise shaping. This new architecture allows for a 6x increase in loop BW when similar phase noise is considered; or a 24dB improvement in phase noise when the bandwidth is kept the same.

Chapter 5: Spur Tone Calibration

5.1 Origin of Spurs

We have proposed a new fractional-N PLL architecture where a sinusoidal reference is used to generate multiple edges per cycle for faster phase update. This acts like a reference multiplier and allows for better quantization noise shaping. The non-uniform clock derived from the sinusoid has the longest repeating pattern at f_{ref}, leading to FM sideband at $f_{ref,}$ even though the update rate is 6x higher on average. The origin of the spurs can be explained as follows. Figure 5.1 shows a few cycles of the irregular reference clock and the divided clock. In steady state, the phase difference between the irregular reference and the similarly divided down clock produces a ripple on Vtune which is proportional to the phase error Φ_i. We approximate the nonlinear sampled data system with linear analysis for small signal response.

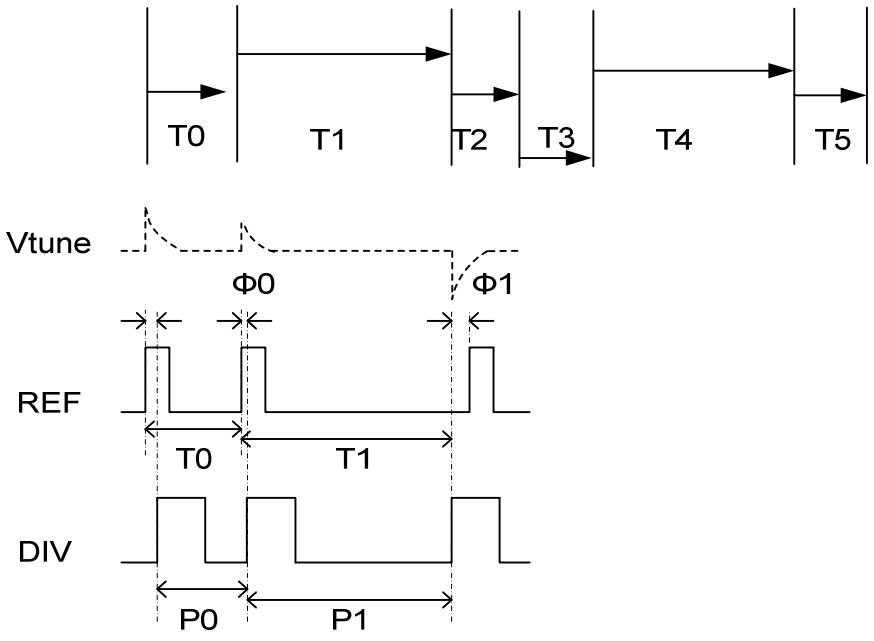


Figure 5.1: Origin of Spurs

Assuming the frequency content of Vtune at f_m is V_{rms}, the modulation side band relative to the carrier at f_m is found as:

$$SB = V_{rms} \cdot \sqrt{2} \cdot K_v \cdot J_1(m)/J_0(m) \quad (5.1)$$

Here the *J*'s are the Bessel functions where J_1 is related to magnitude of the 1st side band and J_0 is for the carrier, m is the modulation index and K_v is the VCO gain [2]. If T_i and P_i are exactly matched as described in 4.3 for $i=0$ *to* 5, then Vtune only accounts for the residual fractional error and occasional noise, where the fractional error is high-pass filtered due to the ΔΣ loop. Simulations showed the total energy of these spurs, referred to as *inherent spurs* hereinafter, is much less compared to integrated phase noise. In actual implementation, the squaring buffers used in the frequency multiplier are subject to process and temperature variations. As a result there are static mismatches between T_i and P_i. This can lead to significant growth in spur energy. We devised a calibration scheme using the Newton-Raphson method [38] to find the optimum divide pattern P_i in the search space, which best match the irregular reference cycle T_i for i=0 to 5, resulting in a calibrated spur energy similar to that of the inherent spurs. The calibration algorithm is described next.

5.2 CALIBRATION ALGORITHM

The calibration algorithm is described in Figure 5.2. A simple FFT routine is used to recursively calculate the total spurious energy in the PLL output clock while perturbing the divider pattern (P_0 to P_5), we move the divide pattern consecutively toward the direction which lowers the spurs. The calibration ends when a global minimum in spurious energy is found or a pre-specified precision is reached. The calibration scheme has been successfully simulated in Matlab, and the results are shown in Figure 5.3.

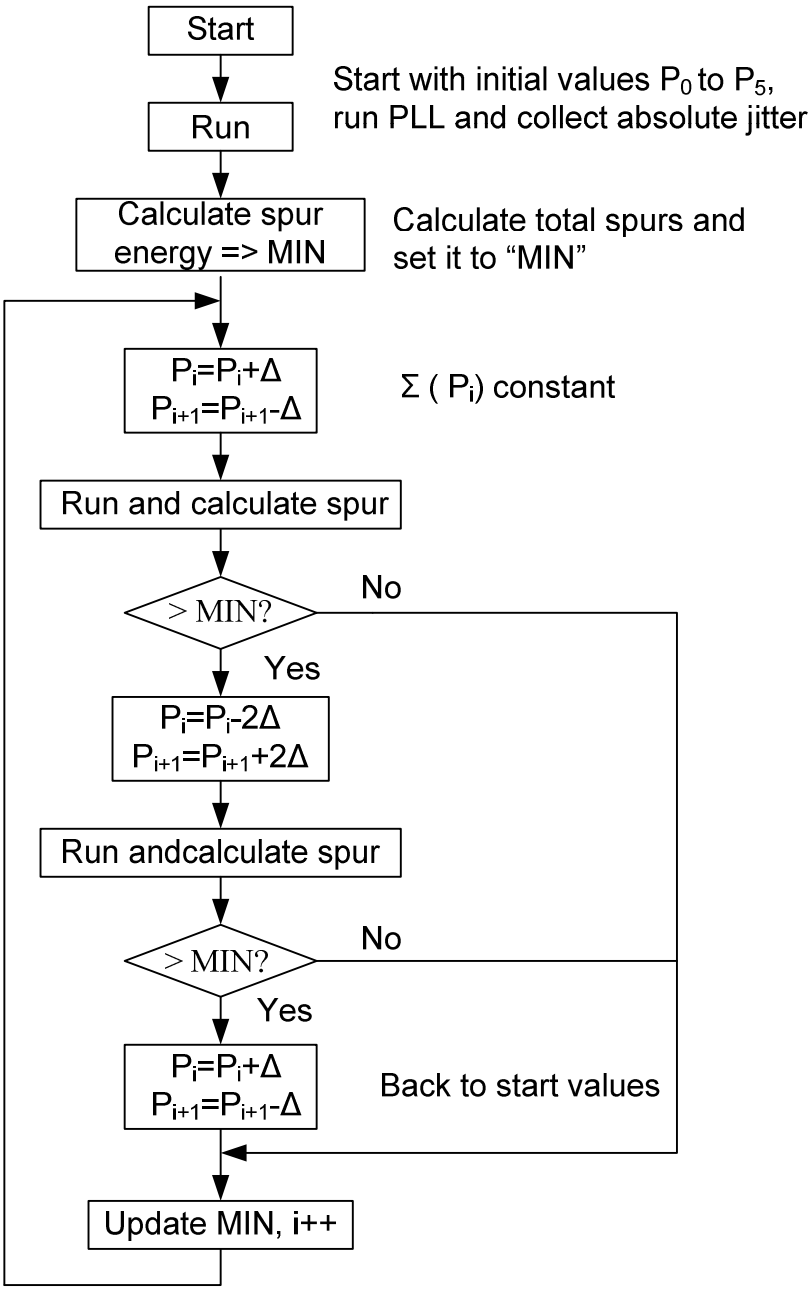


Figure 5.2: Calibration Algorithm

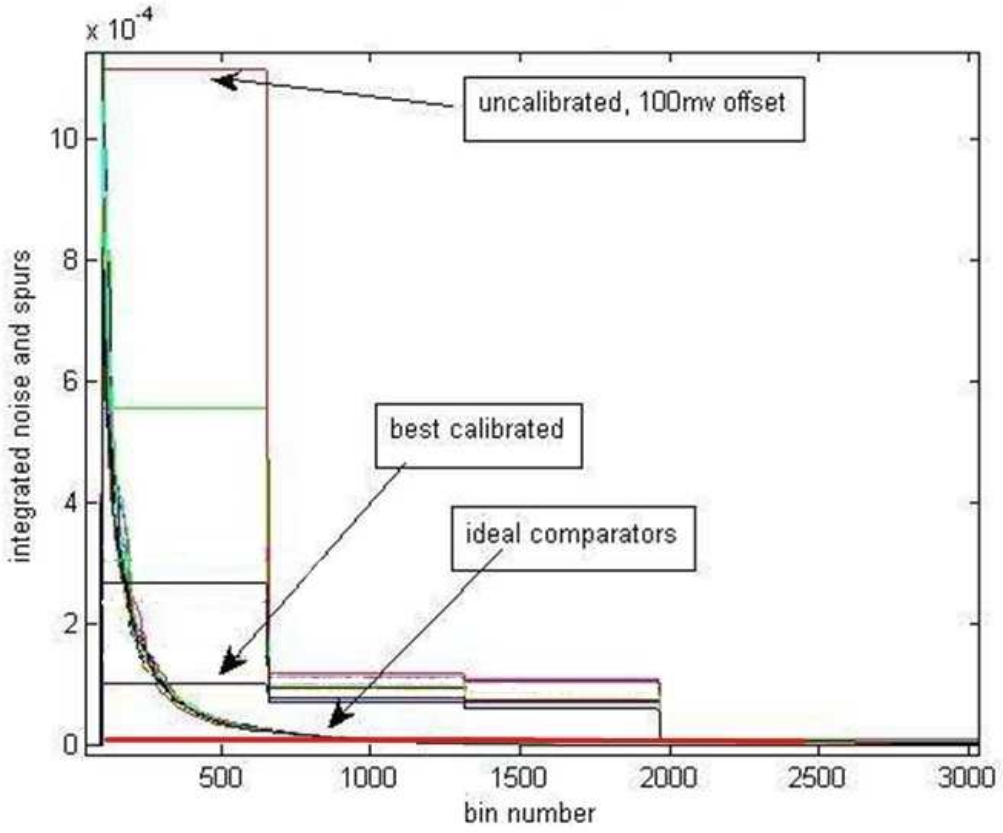


Figure 5.3: Spur Tone Calibration

In Figure 5.3 the stepped curves show spur energies summed from the 6th tone to the 1st tone. The noise is integrated separately for the same frequency range, seen as a cluster of curves. The integration was carried out from the high frequency bin to the low frequency bin to best illustrate the effect. To simulate the calibration scheme, random 10% offsets are introduced into the squaring buffers. The starting point shows the worst case total spurs (red) which dominate the noise power. After 9 iterations the total spur energy was reduced by 10x. The "best calibrated" curve (blue) marked in the plot does not represent the limit of the calibration scheme. In fact by allowing sufficient iterations, the calibrated spurs approach the inherent spurs (cyan). Note that the integrated noise is

similar for all cases. This is also compatible with what we observed in Figure 4.11 and discussed in Section 4.2.

Alternatively, the calibration scheme can be carried out by perturbing the comparator thresholds instead of the divide pattern. The buffers used in the reference multiplier are essentially comparators with built-in thresholds, where the thresholds are determined by the device strengths in the buffers and the degeneration resistors. By systematically perturbing the thresholds, the same calibration result can be achieved.

5.3 IMPROVEMENT ON CALIBRATION

We have developed a calibration algorithm based on PLL output phase. A simple recursive DFT engine can be implemented to compute the spurious energy, which in itself has very low area impact since only a sub-set of frequency bins are involved. However the algorithm requires accurate phase measurement at f_{vco}. A good reference at f_{vco} in an SoC environment is difficult to come by, which makes the calibration scheme unrealistic for certain applications. In this section, an improvement of the calibration scheme is described.

The phase detector measures the timing difference between the input clock and the feedback clock, and produces an error signal which drives the loop to steady state. Under locked condition the error signal is nothing but a sequence of narrow pulses with a net sum of zero. The frequency contents of the error signals are highly correlated with the output phase. Instead of measuring output phase, we can obtain the same information for calibration from the phase detector output, in fact only a running average is needed. In a digital implementation where the error signal is represented as a digital word, the average is easily obtainable through a set of accumulators. In this modified calibration scheme, the phase detector output pulses are summed according to their index position in the XO

period over 1000 cycles, and normalized to the unit interval (UI) by multiplying the average reference frequency, f_{avg}. The result forms an error pattern δ_i, $i=0$ *to* 5. δ_i represent the average phase error which update the loop six times per XO cycle. Ideally the error pattern should be zero once the loop is locked. Any static mismatch between P_i and T_i will lead to a significant increase in δ_i. The calibration algorithm is based on this set of time domain information. A 50 to 100 mV offset is randomly introduced on each T_i. During each calibration run, the max(+) and min(-) phase errors and their indices are located, then a Δ is subtracted in the divide pattern at the maximum index, and compensated in the minimum, allowing the total divide ratio in each XO period to remain constant and equal to N+α on the average. The step size Δ is adaptable depending on the magnitude of the error. The error pattern after 10 calibration runs is reduced over 10x. This is shown in Figure 5.4. Note δ_i are defined on reference edges indexed from 1 to 6, corresponding to the irregular cycle defined by T_i for $i= 0$ to 5.

From the steady state jitter we again calculate phase noise and the results are plotted in Figure 5.5. Only three runs are plotted for picture clarity. At the start of calibration, the PLL exhibits strong spurs at 10MHz and its harmonics. After 10 iterations, the 10MHz spur is reduced by 15dB, the second and third spurs are reduced by about 20dB, while the fourth spur is reduced to below the noise floor. As discussed in Chapter 4, and also shown in Figure 5.5, that the static mismatch between Pi and Ti does not degrade the phase noise in a significant way, it mainly affects spurs.

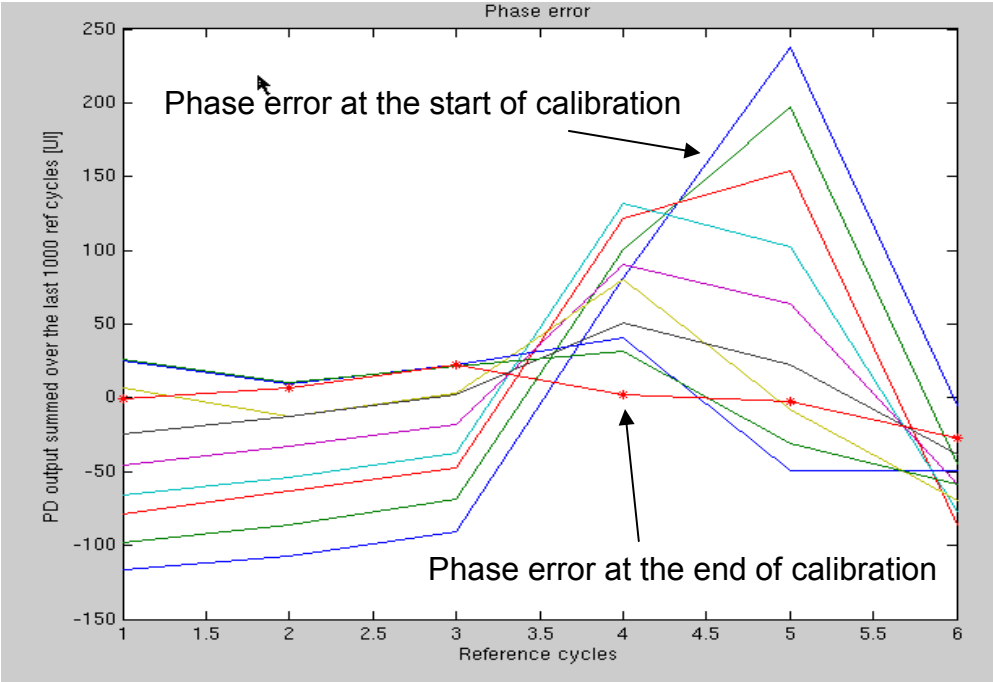


Figure 5.4: Error Pattern over 10 Calibration runs

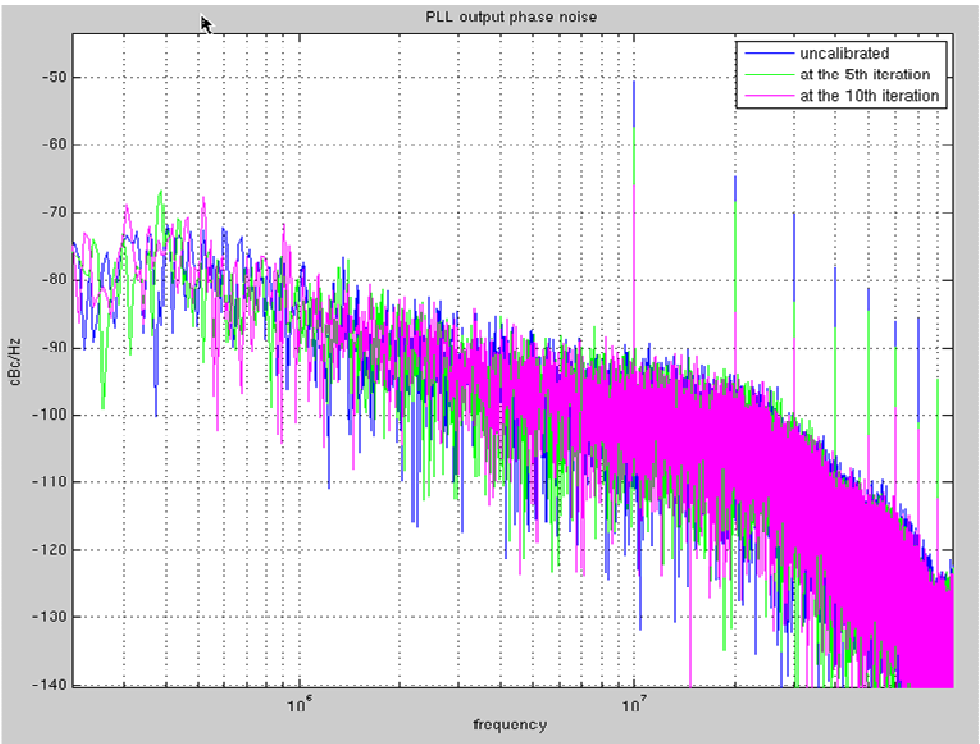


Figure 5.5: Spur reduction through calibration

5.4 SUMMARY

We have developed a new PLL architecture where a fast clock with irregular cycles is used to lock the loop. One problem associated with the proposed approach is the degraded spurious performance due to process and temperature variations. These PVT variations alter the thresholds of squaring buffers and create a mismatch between the reference cycles and divider pattern. In this chapter we presented a calibration algorithm based on Newton-Raphson method to overcome this problem. The calibration scheme calculates the spurious energy and modifies the divide pattern towards a direction which minimizes the spurious energy. An improvement in the calibration scheme is introduced in Section 5.3, in which an error pattern is calculated and used to modify the divide pattern. This makes it possible to implement this technique for $_{\text{applications}}$ when a quality high frequency on-chip reference is unavailable. In simulation, the calibration scheme was shown to lower the spurs down to inherent spurs level, which becomes insignificant compared to the phase noise.

Chapter 6: Circuit Implementations

6.1 Reference Multiplier

The schematic for the reference multiplier is shown in Figure 6.1 with a sinusoidal input "Vin" driving three buffers. The first stage buffers use fairly large devices to reduce its 1/f noise contribution. An N-well resistor was used for source degeneration for two of the buffers to create offsets. The value of the resistor was determined through simulations; 500 Ω was needed to place the edges roughly at ±45°. As described in Chapter 3, the exact angle does not matter, as long as there is sufficient gain at the phase to voltage slope (so as to not to fall in the flat region of the sine wave). In addition the trip points should be reasonably displaced from the zero crossing. The first condition ensures a good phase noise on the derived clock; the latter prevents the loop from overloading when the variations among these irregular cycles become too large. A pulse of 1 ns is formed for each of edge created by the offset buffers and combined into the final clock, "Vout". Only the rising edges are used in phase comparison, careful attention was paid to maintain a sharp transition on this edge through the signal path. The simulated results are shown at the top of Figure 6.2.

This idea of frequency multiplication can be extended to quadrature references for systems where such references are readily available. A poly-phase filter can also be used to generate quadrature signals [39], in which a simple RC network is used to create 0°and 90° phase shift. When quadrature references are used for the proposed frequency multiplier, additional edges can be extracted from the cosine wave when the sine wave flattens out and vise versa. This provides a 12x increase in the reference frequency. The simulated waveforms are shown at the bottom of Figure 6.2.

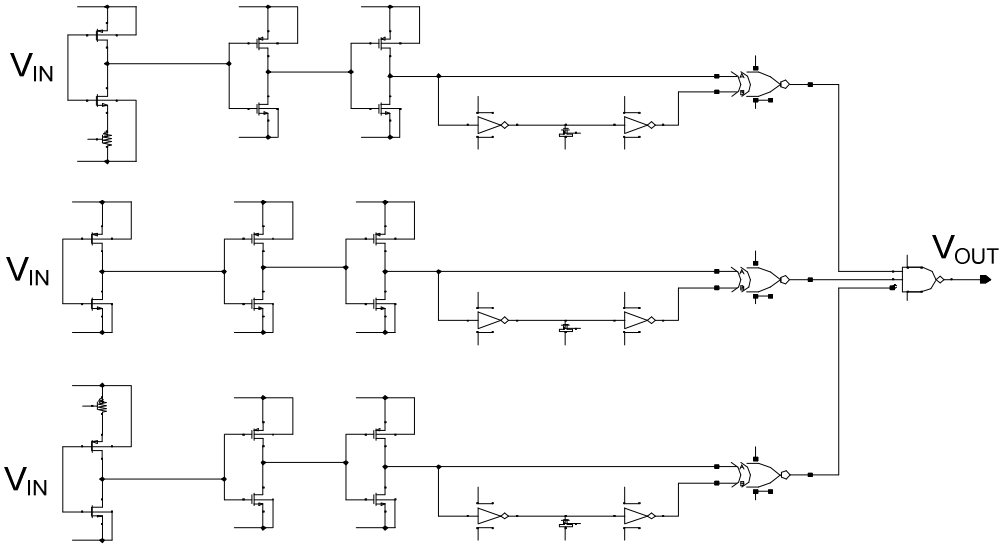


Figure 6.1: Reference Multiplier

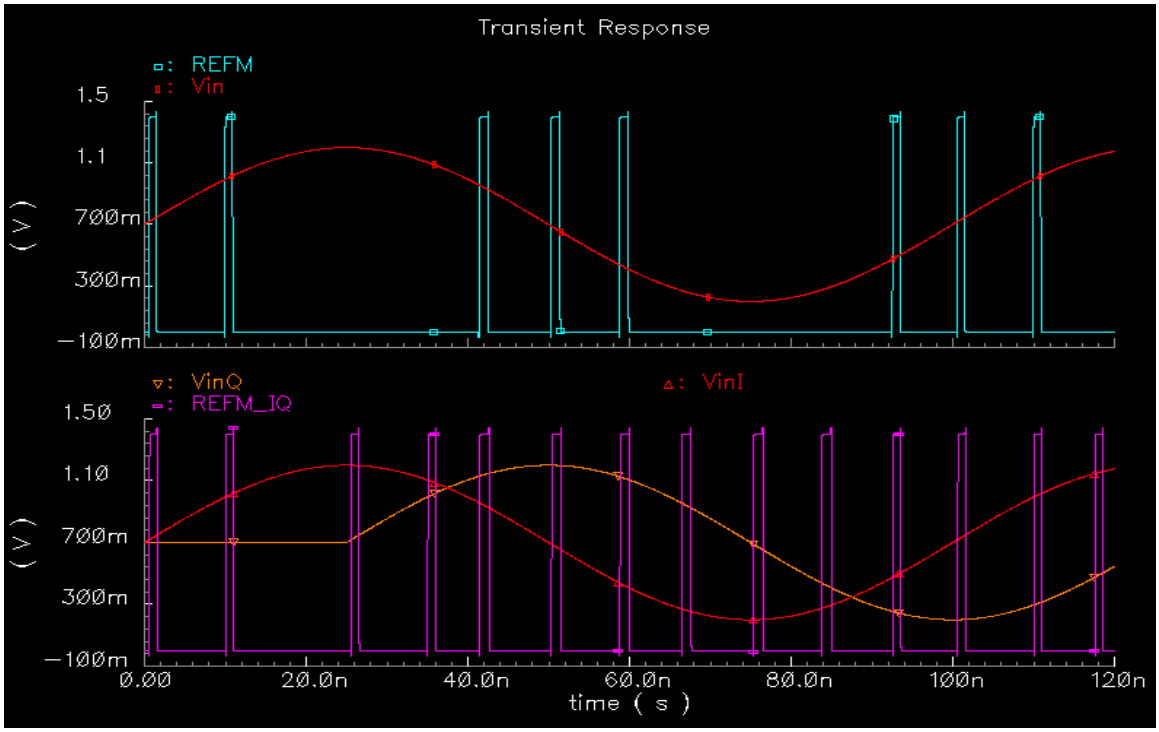


Figure 6.2: Circuit Simulations on REFM

6.2 PHASE DETECTOR

6.2.1 The Mixer PD

There are various ways to implement a phase detector. A mixer can be used as an analog approach for phase detection [2], [3]. In this case two sinusoidal inputs with equal frequencies $A_1 \sin(\omega t + \varphi_1)$ and $A_2 \sin(\omega t + \varphi_2)$ are multiplied in a mixer, the output is given as:

$$V_{PD} = A_d \sin(\varphi_1 - \varphi_2) + A_s \sin(2\omega \cdot t + \varphi_1 + \varphi_2) \qquad (6.1)$$

where the first term is the desired signal at DC and the second is the unwanted signal at twice the reference frequency of the input signal (the sum-product). A signal at the reference frequency can also be present depending on the LO-to-IF and RF-to-IF isolation. This kind of PD can operate at very high frequency and is good for applications where the loop BW is kept low enough to suppress the harmonics.

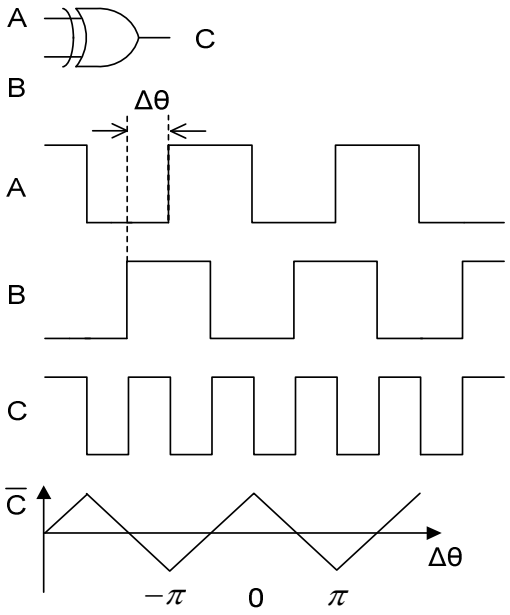


Figure 6.3: EXOR PD

6.2.2 The EXOR PD

A digital equivalent of an analog multiplier is an EXOR gate. Similarly an EXOR gate can be used as a PD. It requires two input clocks with similar frequency. The waveforms are shown in Figure 6.3. When the two inputs with equal frequency are off by ± 90°, the output waveform is at twice the frequency and averages to 0. As one input is phase shifted against the other, the average output is a triangular waveform with its linear operating region stretched from $-\pi$ to 0, or from 0 to π. This limits the PLL locking capability. In addition the EXOR PD would not produce meaningful result if one input is twice the frequency of the other.

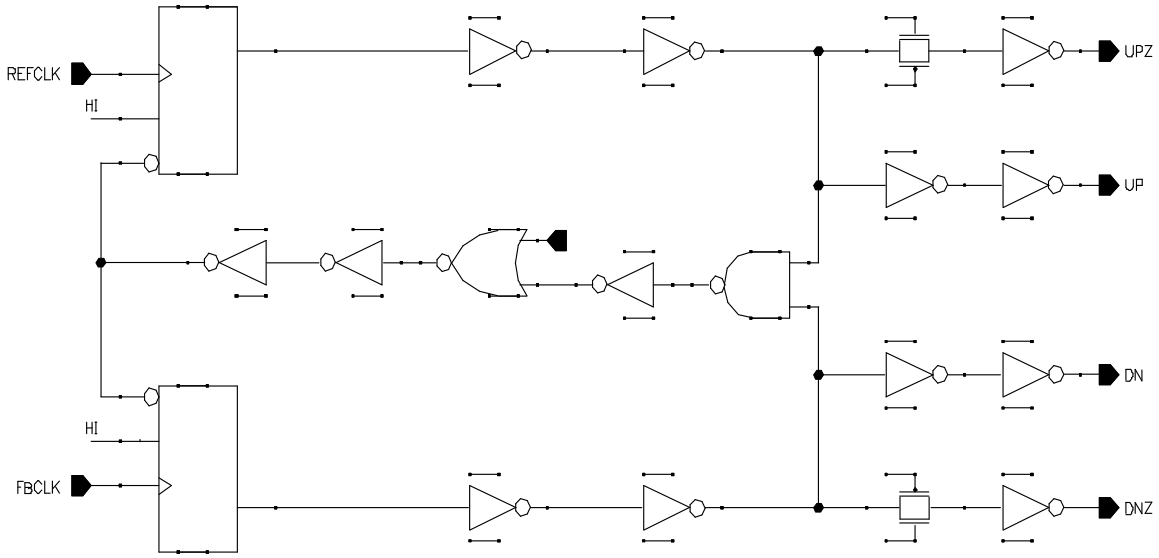


Figure 6.4: Schematic of PFD

6.2.3 The Phase Frequency Detector

A popular PD uses an asynchronous tri-state state machine implemented with two flip-flops. The built-in memory function allows the PD to detect frequency as well as phase. The schematic used for this work is shown in Figure 6.4. A rising edge of the REFCLK triggers an UP pulse which is turned off only at the arrival of the next rising

edge of FBCLK. Similarly a rising edge of the FBCLK triggers a DN pulse which is only turned off by the next arrival of the REFCLK. The gates inserted in the reset path add a fixed delay which will cause an overlap at the end of the UP and DN pulses, where both are momentarily turned on. This removes the dead zone in the phase detector [2]. It is important to keep both the UP and DN pulses with equal path delays. A transmission gate is also used to pad the path between UP and UPZ, and DN and DNZ. In addition mirror symmetry is maintained in the layout from top half to the bottom half in order to minimize skew.

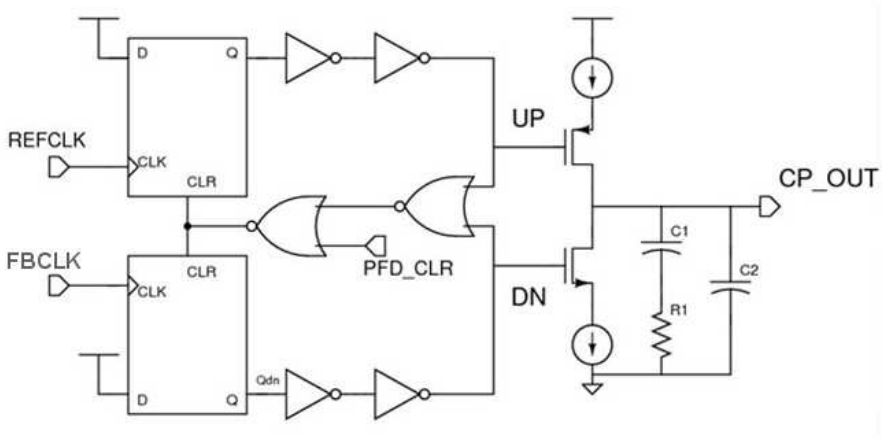


Figure 6.5: PFD combined with CP

The PFD can be conveniently combined with a CP as shown in Figure 6.5. The UP and DN pulses are used to turn on and off a pair of switches that controls a pair of current sources in the CP. With arbitrary clock edges for REFCLK and FBCLK as illustrated in Figure 6.6, the UP and DN pulses and the charge packets delivered by the CP to loop filter is also shown in the same plot. Note there is a brief overlapping period when both UP and DN are on. By sweeping one input clock against the other (both with equal frequency), we can see the linear range extends to from -2π to 2π.

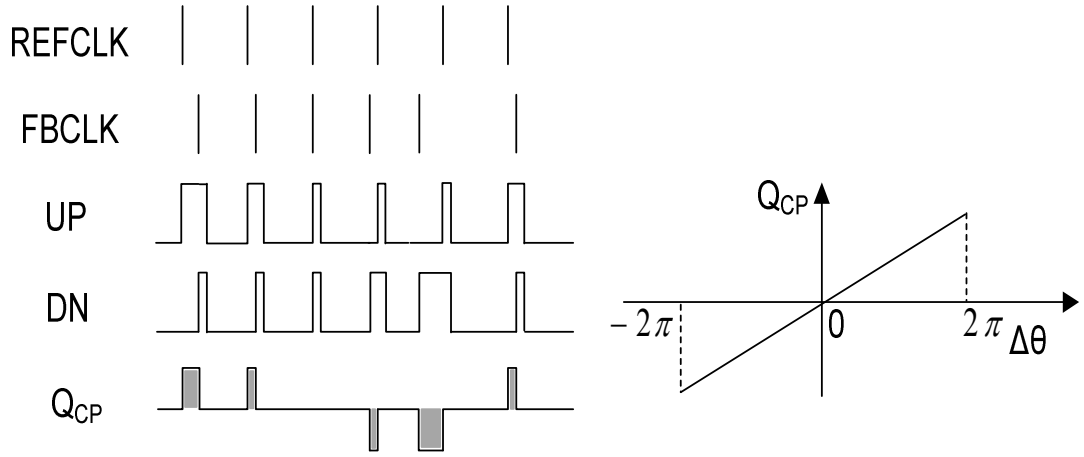


Figure 6.6: PFD Linear Operation Range

6.3 CHARGE PUMP

A current steering CP was selected for its fast switching time [48]. The simplified schematic is shown in Figure 6.7. The switches are set in the ohmic region driven by full swing UP and DN pulses. This is to avoid the fast switching of bias signals as needed for switches in saturation. Overlapping clocks were created between the complimentary UP and UPZ, DN and DNZ pairs to ensure a continuous flow of current. The cascode devices provide a good isolation between the CP plus and minus nodes [11]. In addition, a buffer is used to balance the CP output with the dummy branch, which prevents charge sharing from the dummy side. It is important to keep a linear conversion gain from the phase delta and the charges delivered by the CP. Nonlinearity in this transfer is particularly harmful in a wide band fractional-N synthesizer, since this could result in quantization noise folding down from the high frequency. A fully differential CP design is recommended wherever possible. In this test chip, we used a single ended charge pump in

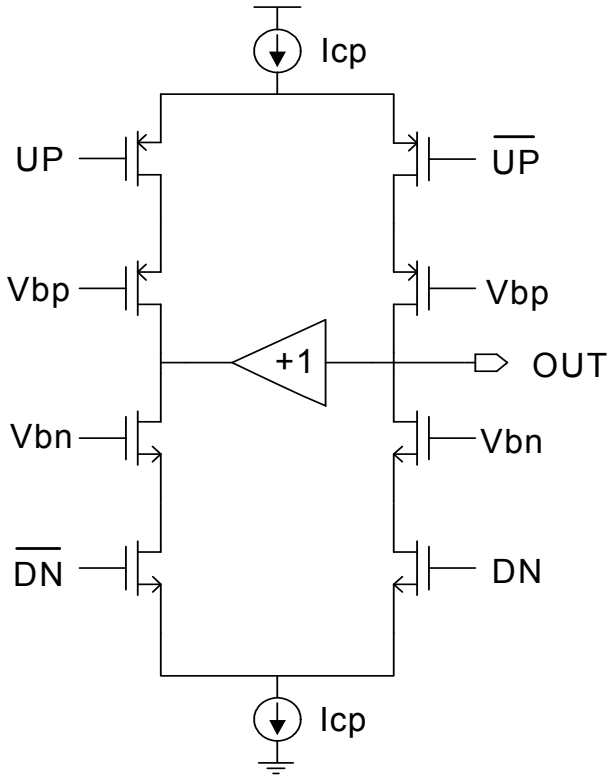


Figure 6.7: Schematic of CP

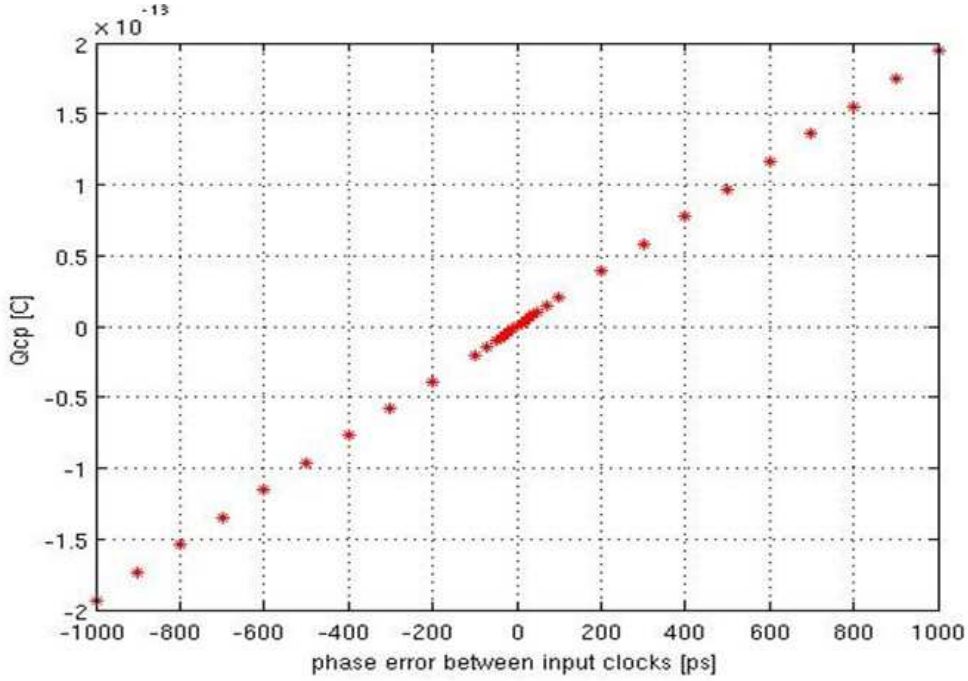


Figure 6.8: Simulated CP Linearity

order to reduce the complexity and lower the risk. The phase to charge transfer is shown in Figure 6.8. There is a small cross-over distortion, which is typical for this kind of phase detection, since multiple current sources are working simultaneously to determine the phase to charge gain. Additionally the two input signals can easily have crosstalk through the power supply [11].

6.4 LOOP FILTER

6.4.1 Active Loop Filter with Explicit CP

A second order active loop filter is used to implement a type II third order PLL [2]. This is shown in Figure 6.9. Compared to the passive loop filter in Figure2.2, the active one

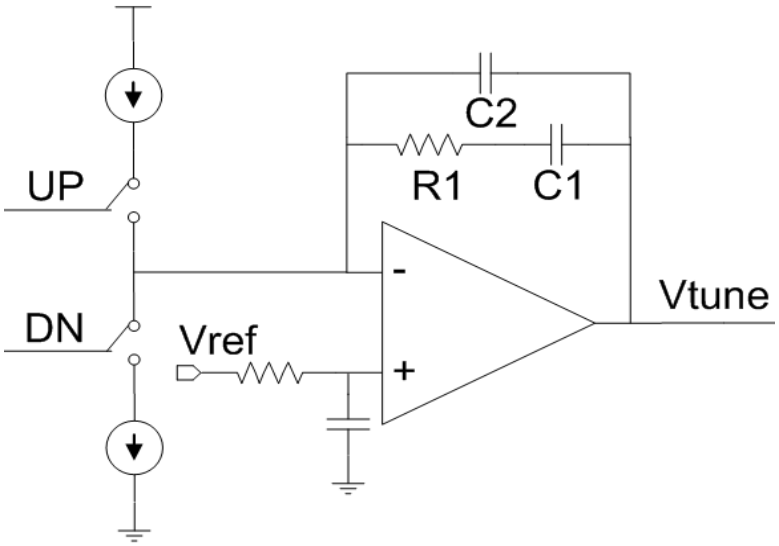


Figure 6.9: Active Loop Filter

provides additional isolation for the Vtune node, and is in general more advantageous for spurious side bands. Both passive and active loop filters give identical transfer function. A well designed OPAMP should contribute minimum noise. The loop filter components

are determined based on the linearized phase domain model described in Section 2.1. The loop gain for the PLL is re-written here as:

$$G(s) = \frac{I_{cp} K_v}{N} \frac{1}{S^2 C_1} \frac{1 + SR_1 C_1}{1 + SR_1 C_2} \tag{6.2}$$

Here N represents the divide ratio between the output frequency and the average frequency of the irregular clock. The compensating zero is located at $1/R_1 C_1$, and the high frequency pole at $1/R_1 C_2$. Additionally the unity gain frequency is approximately:

$$\omega_u \approx I_{cp} K_v R_1 / N \tag{6.3}$$

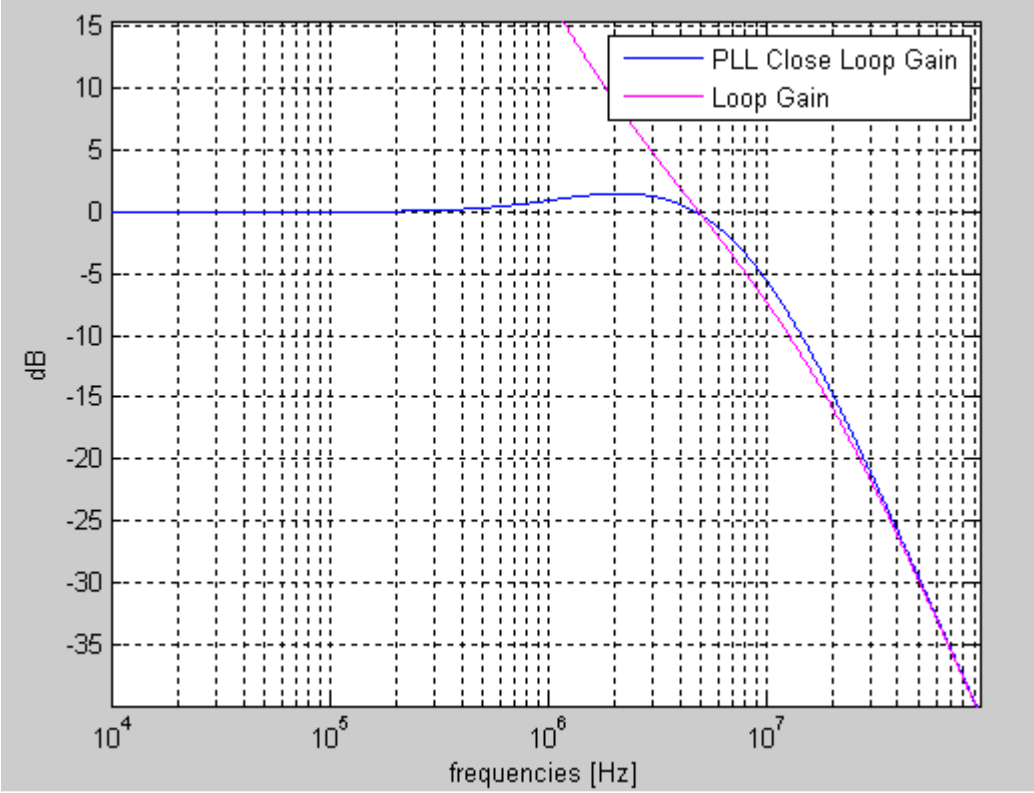


Figure 6.10: PLL Loop Responses

Assume that the output frequency is 2.5GHz, and the input XO frequency is 10MHz. Based on the proposed reference multiplier we have an update rate which averages to

60MHz. In addition from Spice simulation, we find the VCO gain K_v to be approximately 7GHz/V. By setting the unity gain frequency to 6MHz, we then find the loop filter components as follows: Icp=100μA, R1=2KΩ, C1=70pF and C2=5pF. This also puts the zero at 1.1MHz, and the pole at 17MHz. Figure 6.10 shows the loop response.

6.4.2 Active Loop Filter with Implicit CP

Alternatively the loop filter can be implemented using an implicit CP [59]. This is shown in Figure 6.11. This architecture was used in a 32 kHz reference PLL targeted for

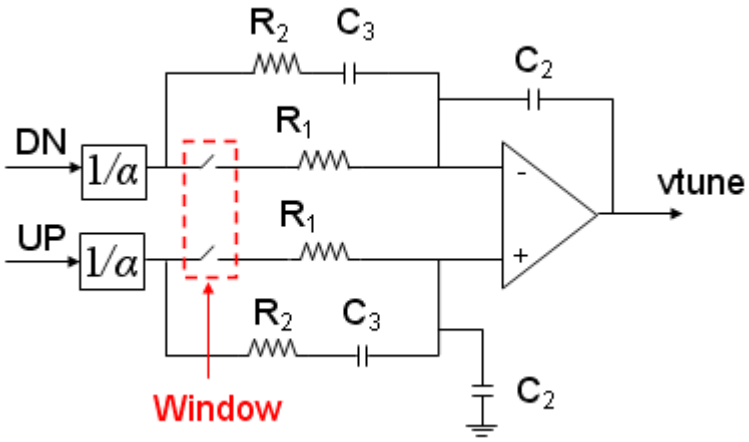


Figure 6.11: Alternative active loop filter

wireless connectivity and broadcast applications. While this architecture is functionally equivalent to the LPF in Figure 6.9, it allows for significant area savings by using a large R and moderate C for loop stabilization. The 'windowing' concept was used to mitigate the problems of excessive thermal noise from the loop filter, as well as spurs from droop in the VCO control voltage due to leakage.

6.4.3 OPAMP

A simple two stage OPAMP was used in the design. It features a rail-to-rail folded cascode input stage and a class AB output stage [40], [41] [59]. The opamp works off a 1.4V analog supply. The schematic without the main biasing block is shown in Figure 6.12.

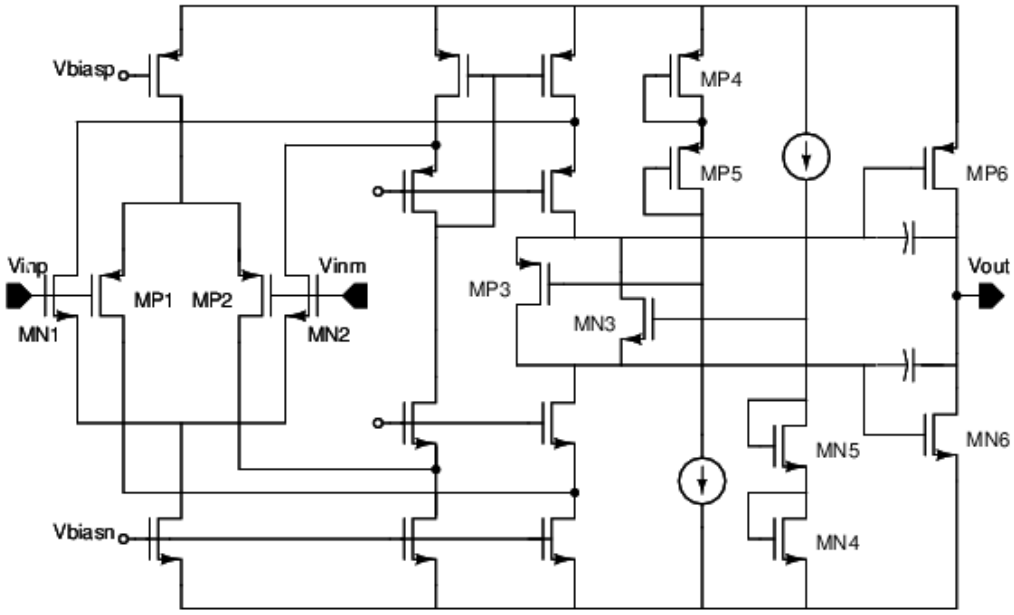


Figure 6.12: OPAMP Schematics

The PMOS differential pair MP1 and MP2 form the main input pair. A small NMOS differential pair MN1 and MN2 is included in parallel, to provide amplification just in case the amplifier input common mode gets stuck at a high value during start up. MP3 and MN3 form the floating-gate battery and are used to enable the class AB operation for power efficiency. In-phase current generated from the differential stage flows into the source of MP3 and MN3, and the two trans-linear loops formed by MP3 MP4 and MP5 and by MN3, MN4 and MN5 are used to bias the output devices MP6 and MN6,

respectively. The output stage was designed to have a strong class A and a weak class B operation, this is to accommodate the loop during transients where a surge of current may be needed due to loop peaking. Figure 6.13 shows the simulation results for the OPAMP output noise, loop response and the filter function. The OPAMP was designed to have a BW that is approximately 10x the PLL BW. The OPAMP noise is dominated by the input differential pair (MP1 and MP) and the summing current sources.

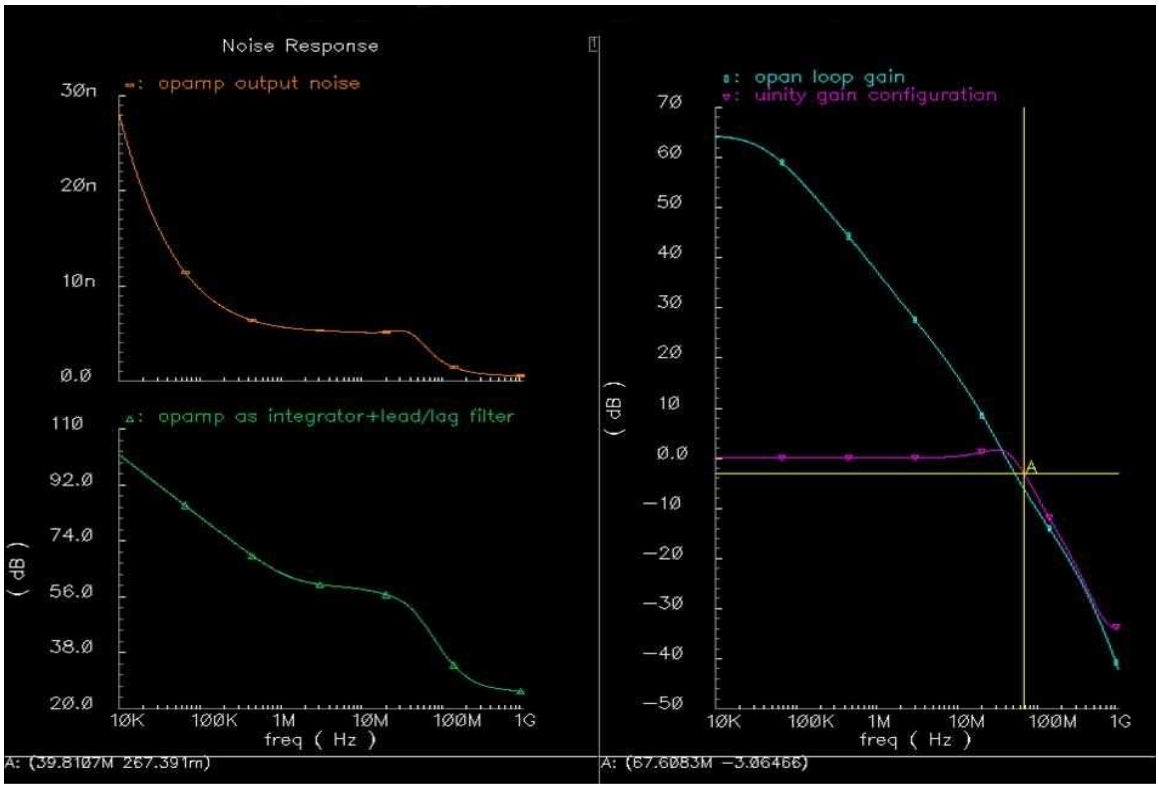


Figure 6.13: OPAMP Response

6.5 VCO

Ring oscillators are widely used in PLLs for their simplicity and smaller silicon area. The oscillator core consists of a chain of CMOS inverters forming a delay line. The frequency of oscillation is determined by the total phase shift of 360° at that frequency.

Note that half the cycle delay is obtained through polarity inversion by using an odd number of inverters. Tuning is achieved by changing the current to charge the inverter paracitic capacitance, and this leads to a large tuning range. The need for using an odd number of inverters limits the highest operating frequency achievable, since a minimum of three inverters are needed. For differential inverters with cross-coupled connections two inverter implementations are possible. A dual-delay path topology has been reported to increase the operating frequency and tuning range [42], in which a negative skewed delay is used in combination with the normal delay to enhance frequency. In spite of its simplicity and suitability for CMOS technology, ring oscillators suffer from poor phase noise due to their inferior Q factor. The Q factor for ring oscillator is given in [2], where n is the number of inverter stages.

$$Q = \omega_{osc} \frac{T_{delay}}{2} = \pi \cdot f_{osc} \cdot T_{delay} = \pi \cdot n \tag{6.3}$$

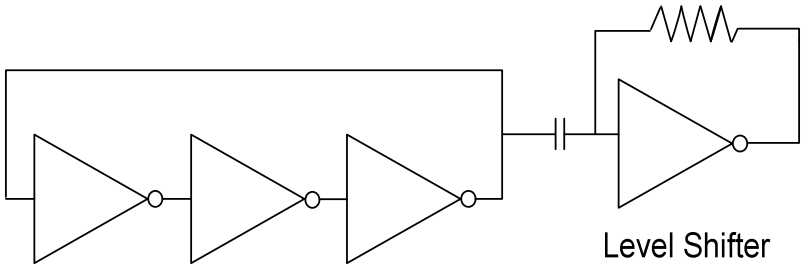


Figure 6.14: Ring Oscillator

The symbolic view of the VCO schematic is shown in Figure 6.14. A simple three stage inverter loop is used in the VCO design. The sizes of the transistors forming the ring which have a strong bearing on the phase noise and power consumption of the VCO have been optimized carefully [49], [50]. The ring oscillator supply is connected to the loop filter output. A level shifting buffer translates the oscillator output to the digital

supply voltage domain. The level shifter is implemented using capacitive coupled self-biased inverter stages. The capacitive coupling places the ring-oscillator output symmetrically around the threshold voltage of the level-shifters, resulting in an excellent duty cycle.

6.6 PROGRAMMABLE DIVIDER

On a cycle by cycle basis, the divider reads in the control bits from the ΔΣ output and divides the VCO frequency accordingly. Typically the divider needs to maintain proper operation at a frequency 30% higher than the VCO frequency to ensure lock acquisition. The divider uses a cascade of divide by 2/3 cells followed by a counter [44], [45]. The divide by 2/3 cell normally divides by 2 unless both the control input and bit input are logic 1, in which case it divides by 3. The schematic is shown in Figure 6.15.

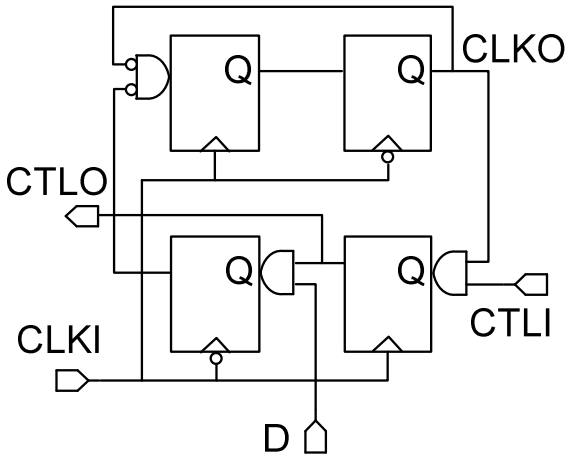


Figure 6.15: Divide by 2/3 Cell

Current model logic is used to implement the latches in these cells, which also have a built-in AND function. The latch architecture is shown in Figure 6.16. The NMOS devices on the left establish a current commuting path, while the cross-coupled NMOS devices provides a positive feedback. The two PMOS devices act like nonlinear

resistances [45], [46]. The divider can divide arbitrarily from 16 to 255. Including layout extracted paracitic capacitance the divider can operate up to 9GHz in simulation. The overall divider schematic is shown in Figure 6.17.

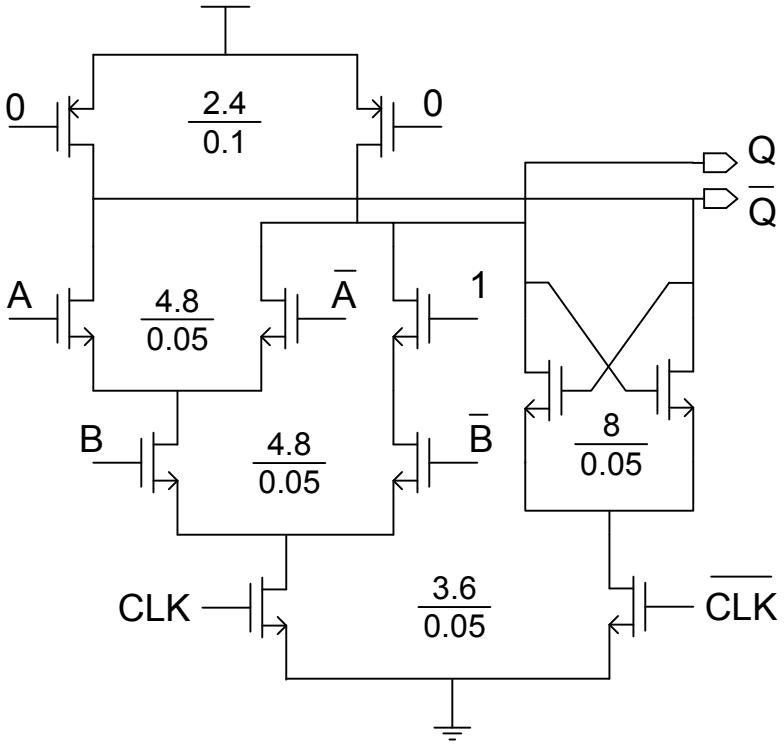


Figure 6.16: CML Latch embedded with AND Logic

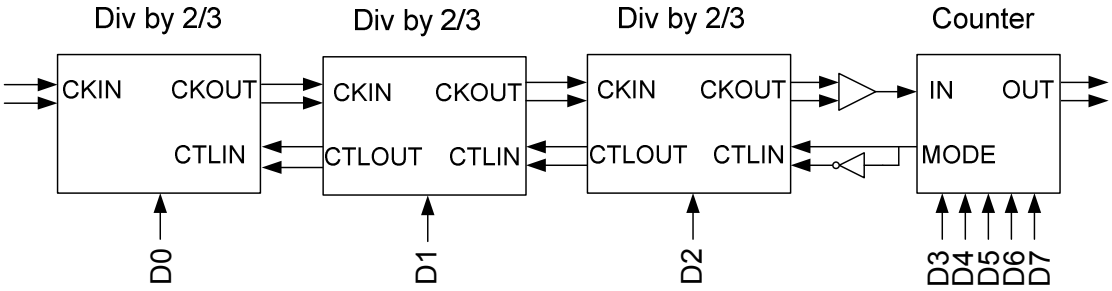


Figure 6.17: Programmable Divider

6.7 ΔΣ MODULATOR AND DIVIDER PATTERN

The digital implementation of the ΔΣ is shown in Figure 6.18 [4]. The input is a 16-bit word representing the desired fractional number α. Dithering is added to the input using a 7 bit LFSR engine. Internal state variables are kept at 19 bits to avoid overflow.

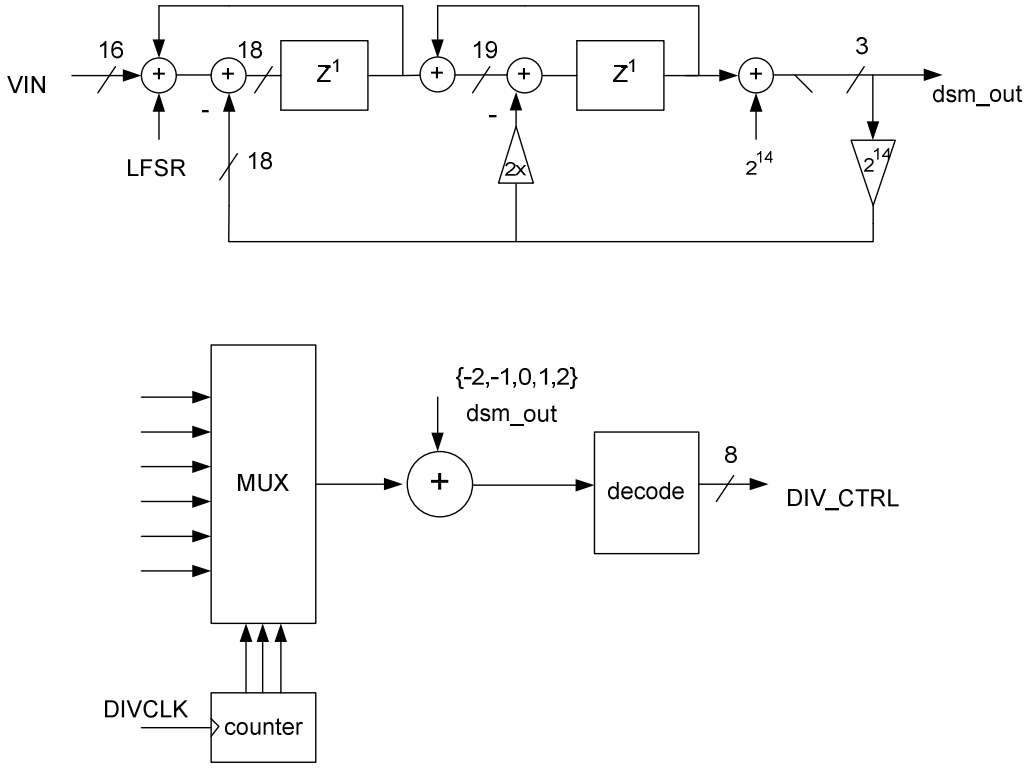


Figure 6.18: Divide Pattern Generation

Outputs are truncated to 3 bit words representing the integer sequence *y[n]* ranging from -2 to 2. The spectrum for *y[n]* is shown in Figure 6.19. The divider pattern Pi, for i=0 to 5, is read in from registers and a counter clocked by the divided clock is used to cyclically select through the pattern. In the end this is summed with *y[n]* and used to control the divider after proper decode. This block is designed in Verilog and synthesized using Synopsys DC compiler.

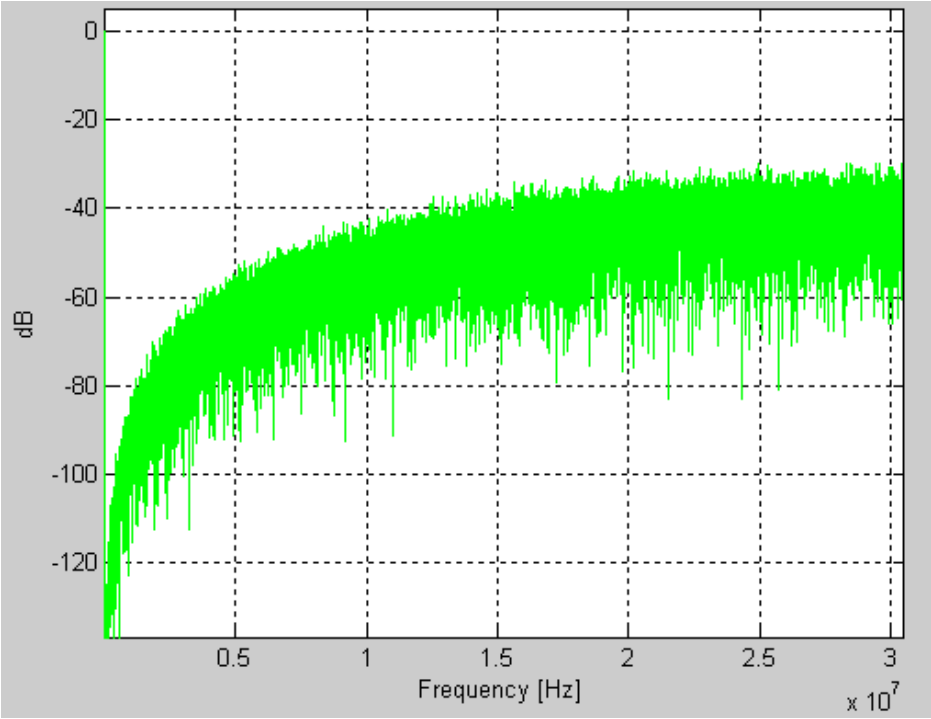


Figure 6.19: DFT of the ΔΣ modulator output

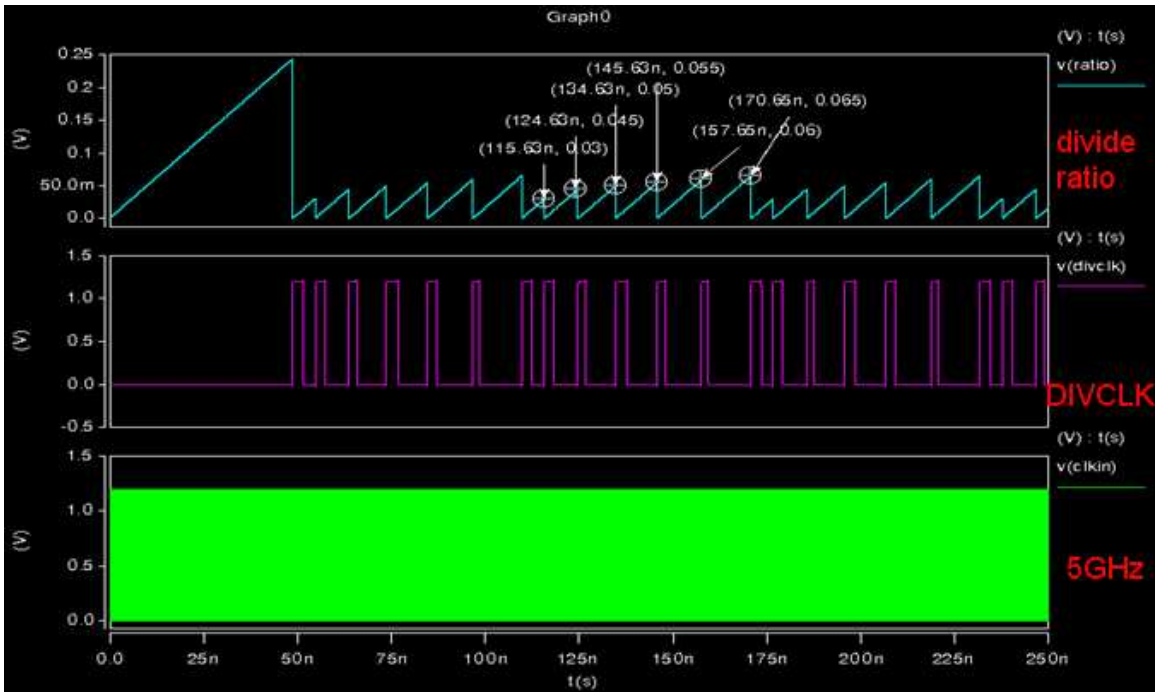


Figure 6.20: Simulation of the Divider with a Control Pattern

6.8 CIRCUIT SIMULATIONS

Circuit level simulations were carried out in Spice or Spectre whenever possible. Mixed domain simulators like NanoSim were also used when Spice level simulations were prohibitively long. For example, NanoSim was used for divider simulations and for closed loop PLL simulations. In this section, we summarize some of the simulation results.

Figure 6.20 shows the simulation where the divider control is set to cycle through six set of numbers 30, 45, 50, 55, 60 and 65. The first waveform is the measured divide ratio. It shows that the control and divider are functioning correctly. In Figure 6.21, the divider was simulated with 10GHz clock and a divider control ratio of 45. The simulation was done in Spice. The first stage divide by 2/3 cell output is also shown. The signal swing in the current-mode-logic cell is about 900mV.

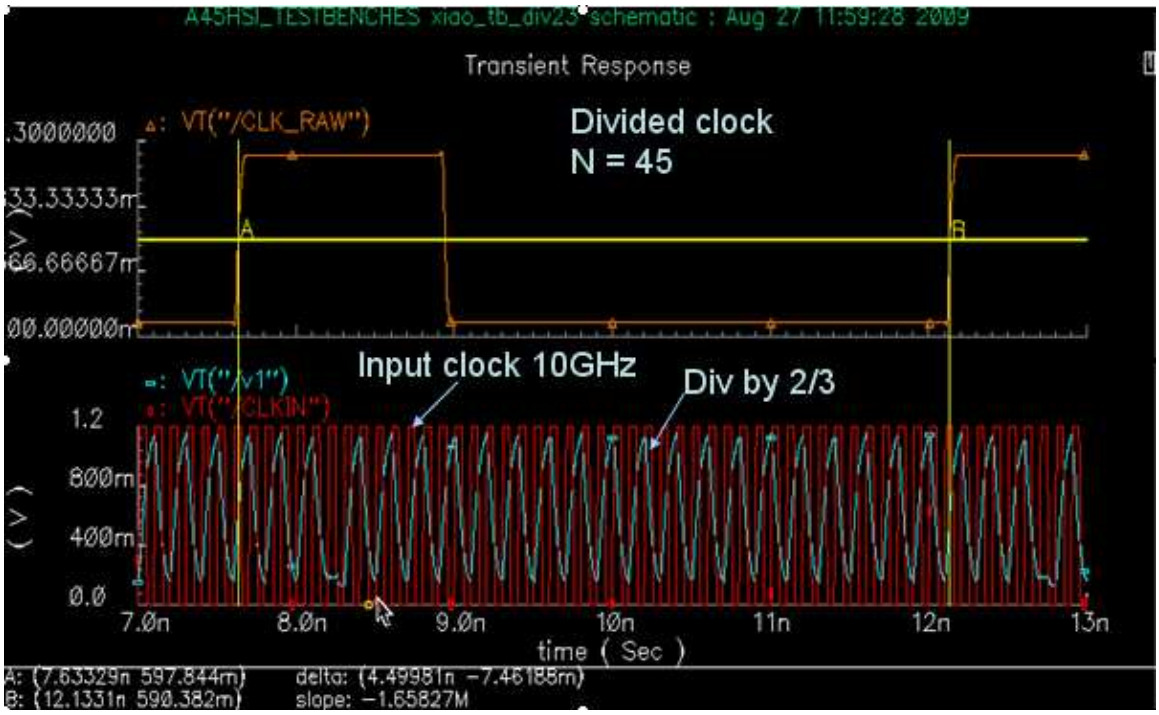


Figure 6.21: The operation of the Divide by 2/3 cell.

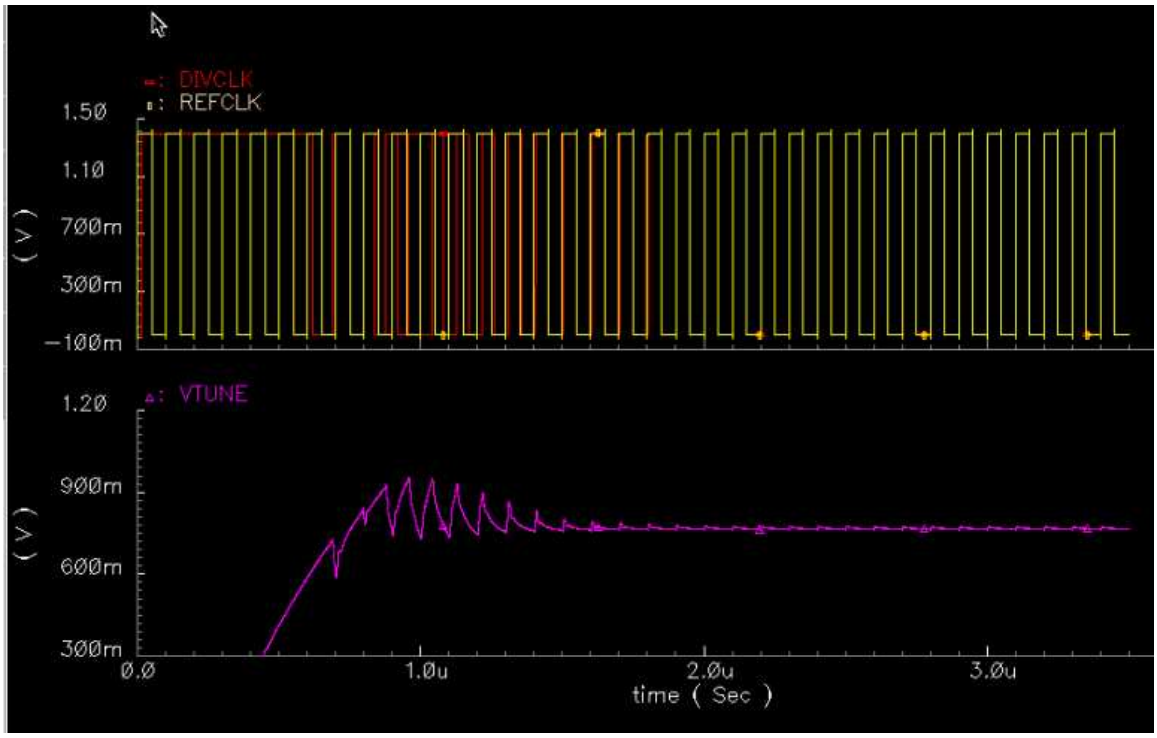


Figure 6.22: PLL Closed Loop Simulation

Figure 6.22 shows the closed loop simulation using Spectre. In this simulation, a Verilog model for the divider was used in order to speed up the simulation. All other blocks are actual circuits. The loop simulation was set in base mode with a 10MHz reference and a constant divide ratio of 300.

Figure 6.23 shows the closed loop simulation in NanoSim [47] where the proposed reference multiplier was used and the loop was locked with an irregular reference which is derived from the reference multiplier. The top plot is the divided clock where a predetermined divider pattern was used. The middle plot is the XO sine reference and the output of the reference multiplier. The reference multiplier produces an internal reference which is on average 6x higher in frequency compared with the XO signal. The zoomed in view is shown in Figure 6.24.

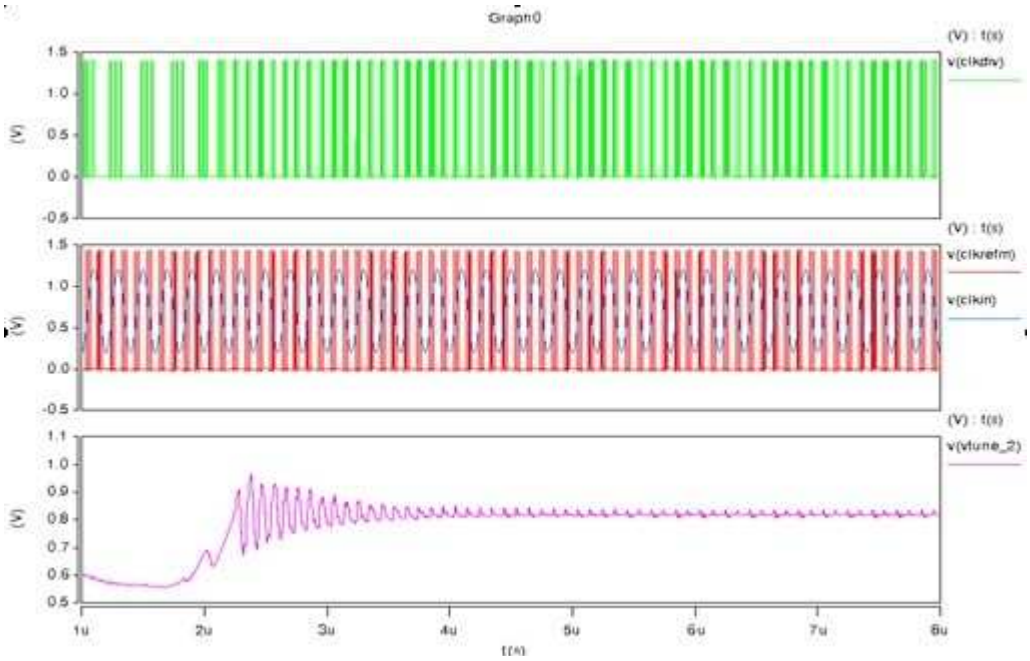


Figure 6.23: Loop Locked with an Irregular Reference

The PLL was taped out in a CMOS 45nm process and the overall layout is shown in Figure 6.25. The total area is 370x290um^2. The PLL shares the pad frame with other IPs. The XO signal is coming from the left hand side and gets multiplied in the reference multiplier "REFM", and it is used as the internal reference and feeds the PFD. The divided clock is routed from the bottom side and also feeds in the PFD. The UP and DN pulses are used to drive the CP which is to the right of the PFD, along with the biasing circuitry. The OPAMP is placed below the VCO. The VCO, being the fastest block, is surrounded by de-coupling capacitances. The divider is placed far away from the VCO to avoid any potential substrate coupling. The ΔΣ and the divider pattern control logic are synthesized blocks and placed next to the divider. The chip micrograph is shown in Figure 6.26. Due to the dummy metal used we are not able to see much detail in the picture.

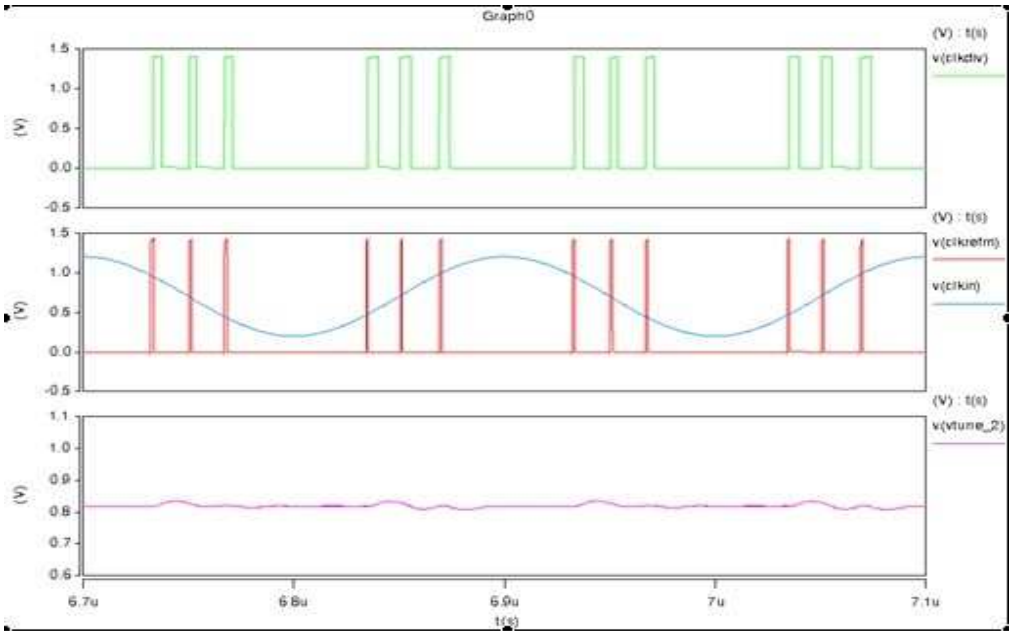


Figure 6.24: Loop Locked with an Irregular Reference (zoomed in)

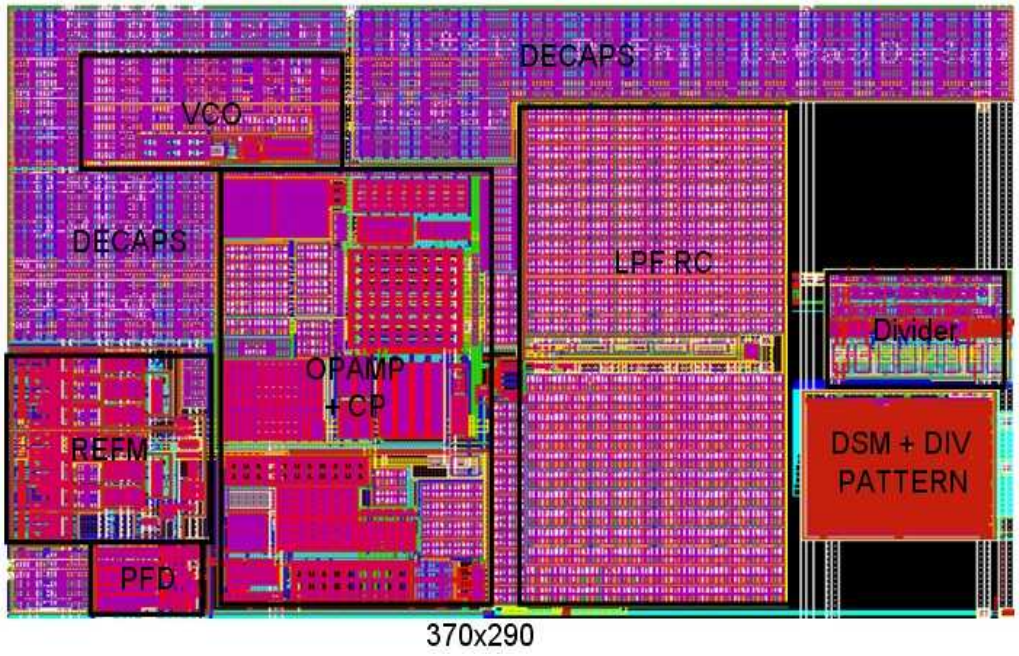


Figure 6.25: PLL Layout

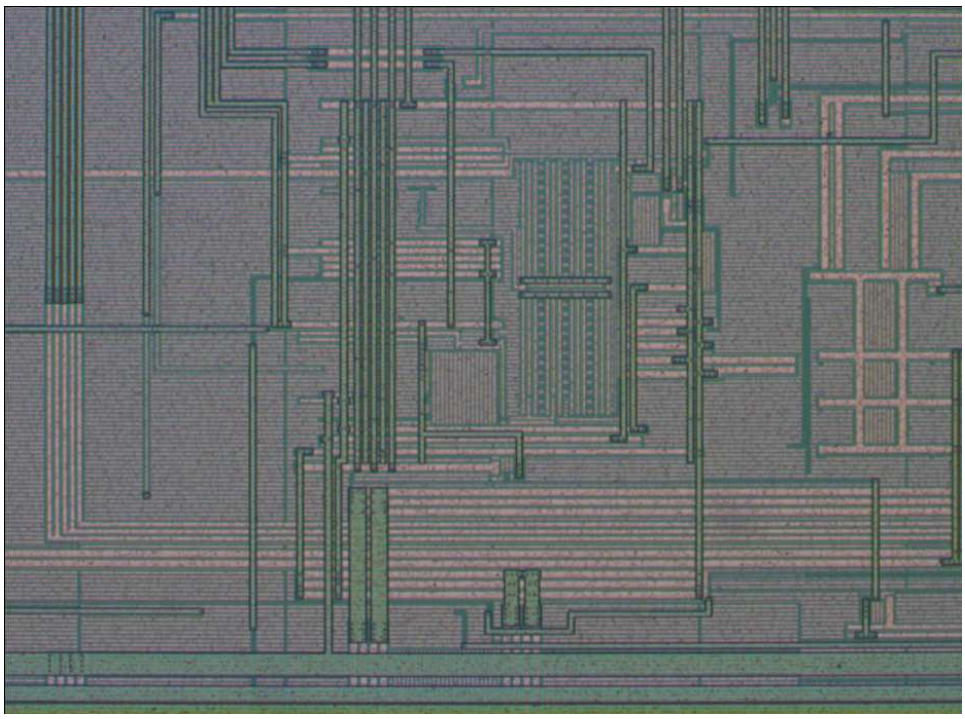


Figure 6.26: Chip Micrograph

6.9 SUMMARY

The implementation details of the proposed PLL are described. The PLL was designed in a 45nm CMOS process and occupies a silicon area of 1.1mm^2. The architecture for each circuit block is carefully selected, implemented and simulated. Circuit and layout techniques are described. We also presented the overall loop simulation results. A mixed signal simulator was used to balance accuracy and simulation time.

Chapter 7: Silicon Results

7.1 Test Setup

The prototype PLL was taped out in 45nm 6 layer metal standard digital CMOS process. The PLL shared a 64-pin pad frame with two other IPs. The test chip is placed inside a BGA (ball grid array) socket which connects to the circuit board. The evaluation circuit board is shown in Figure 7.1. There are five separate power supplies that power the board, the analog and digital blocks, the pad frame, and the output drivers respectively. Large decoupling caps are also used on sensitive supplies. The blue box has a FPGA core which is controlled through a PC user interface to configure the test chip.

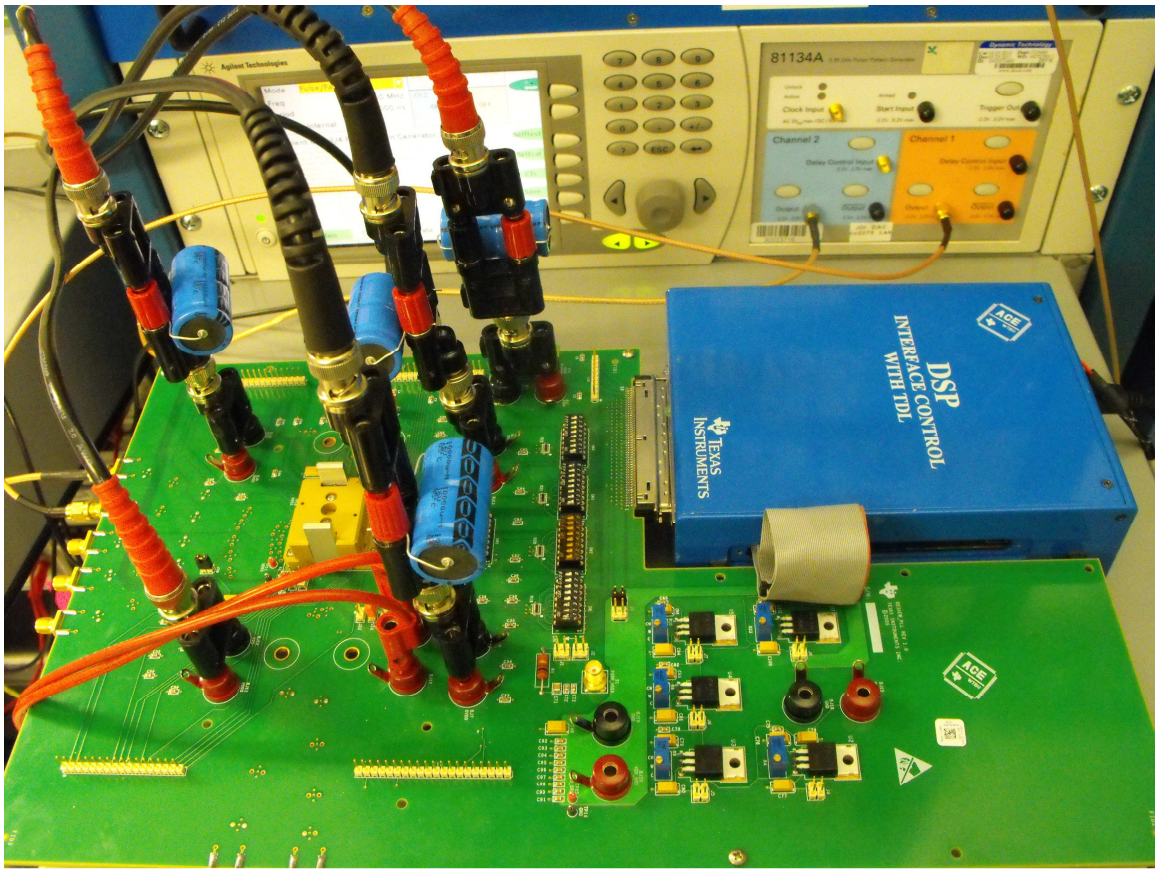


Figure 7.1: Evaluation Board

The blue box allows us to individually power up each of the three IPs which share the test chip, and programme the eight built-in registers. The values in the registers are used to set the PLL operating mode. The input reference is generated from a Rhode and Schwartz SMU 200A signal source and connected to the board through an SMA connector. Alternatively an Agilent 81134 can also be used to provide a clock. Several outputs are observable and taken off the board through SMAs. The equipment used include an Agilent E4443 spectrum analyzer, an Agilent E5052 for phase noise measurements, and a high speed Tektronix oscilloscope to display real time waveforms. The test bench setup is shown in Figure 7.2. With this setup, we program the PLL to operate in the proposed mode and conventional mode and various measurements are taken. Next we will show some of the key measurements and compare them with simulation results.

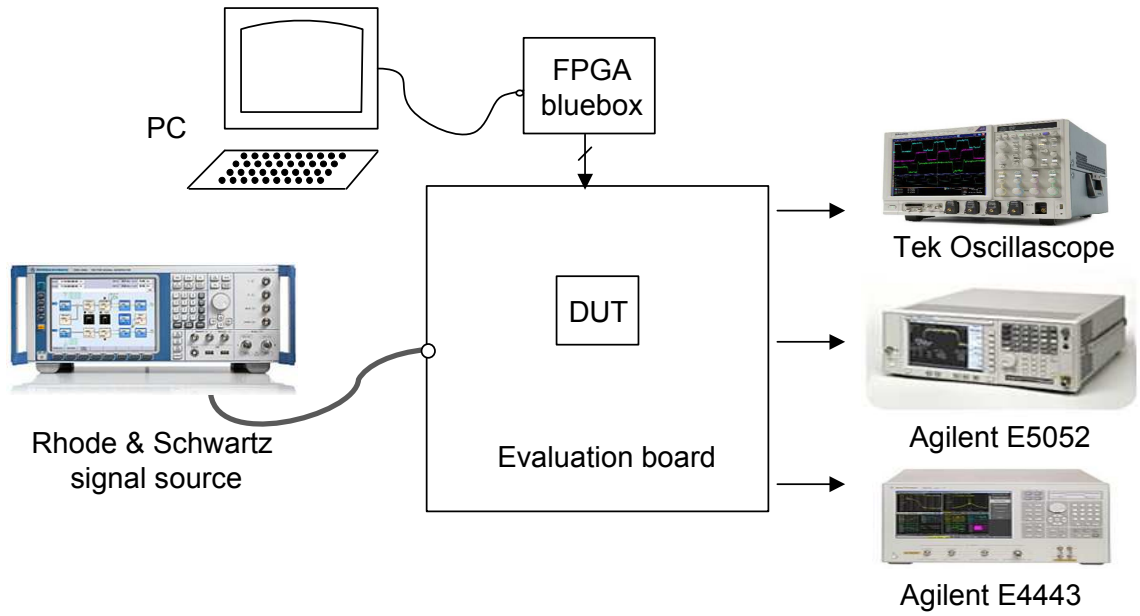


Figure 7.2: Test Bench Setup

Figure 7.3 shows the PLL system as implemented in silicon. The signals for which we have visibility on the bench are highlighted. For ease in referring to various plots that follow, we will establish the following naming convention hereafter:

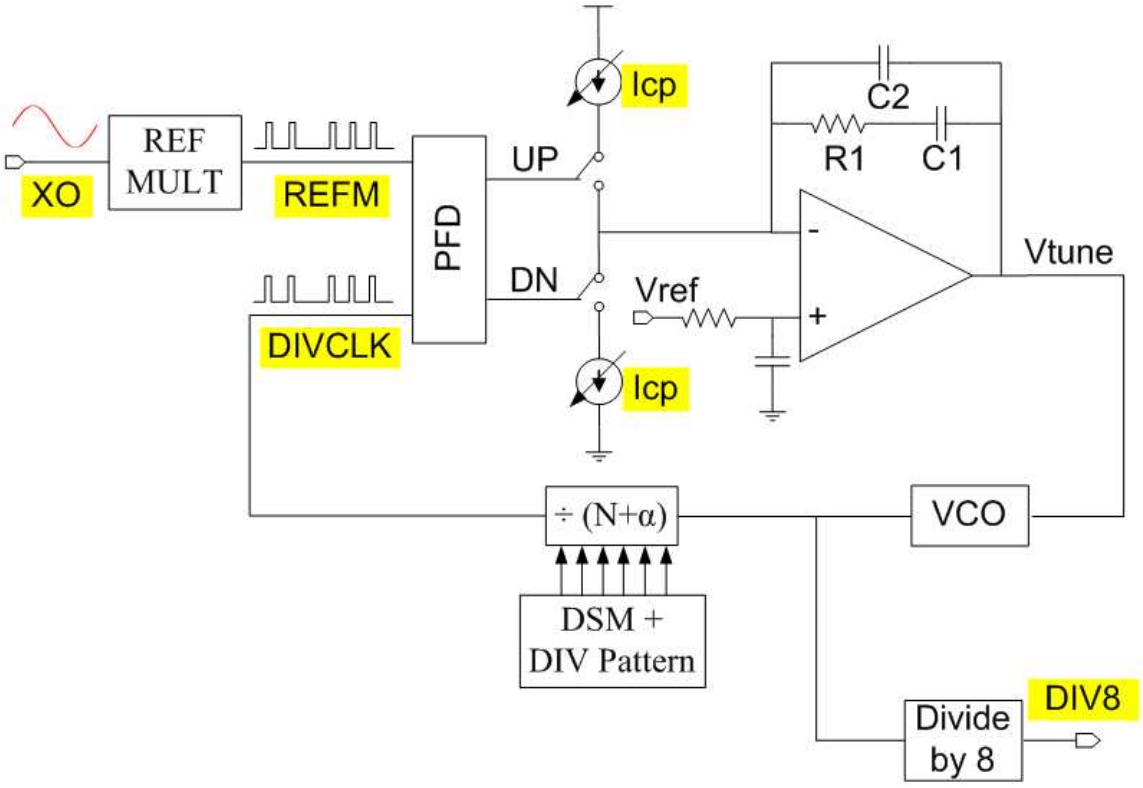


Figure 7.3: The PLL System and the Signals with Visibility

XO – input signal from a sine wave generator.

REFM – output of the reference multiplier; it is also the internal reference.

DIVCLK – feedback clock from the divider.

DIV8 – VCO clock divide by 8.

Icp – charge pump current, programmable to ±50%.

7.2 PLL IN PROPOSED OPERATING MODE

Figure 7.4 shows the measurement results for the PLL operating in the proposed mode. The PLL takes a sinusoidal input XO and multiplies into an irregular clock REFM. Simultaneously a divide pattern is applied to the divider control. In particular, XO is set to 4MHz and magnitude is adjusted such that the irregular cycles in REFM are reasonably distributed. This means the clusters of the cycles are not too close and the variations among the cycles are within 6x. We set f_{vco}=2.4GHz, the total divide ratio in one XO period is set to 600. Based on REFM we programme the divide pattern P_i (for i=0 to 5) to 53, 191, 46, 49, 209 and 52. Once locked, the loop aligns the divided clock DIVCLK with REFM. The locked waveforms are shown in Figure 7.4. For comparison the simulated result for the same settings is shown in Figure 7.5.

Figure 7.4: Locked Waveforms REFM and DIVCLK with ΔΣ Disabled

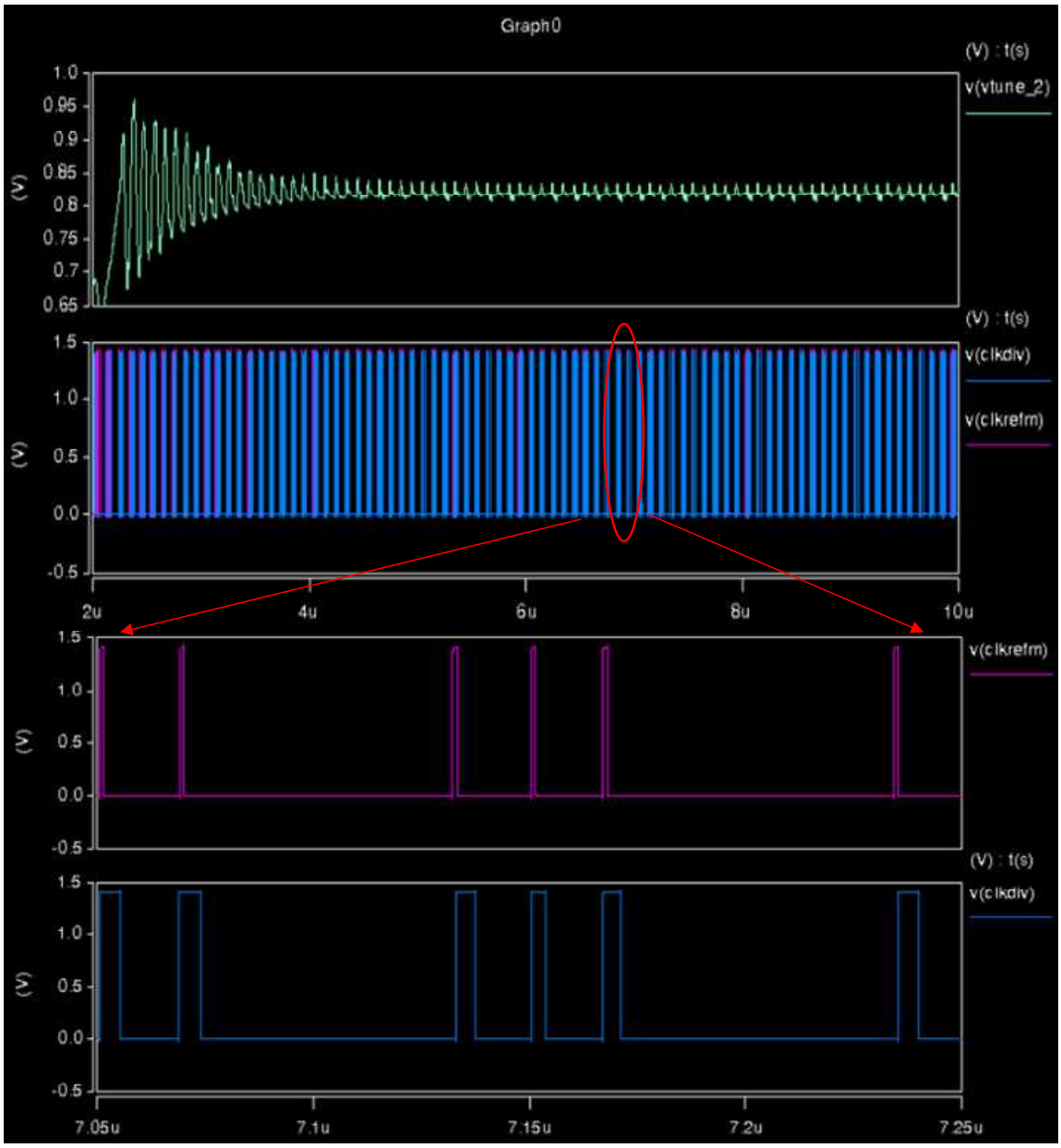


Figure 7.5: Full chip simulation showing locked waveforms of REFM and DIVCLK

Figure 7.6 shows the same set of measured waveforms but with fractional mode enabled. The 16bit fractional word used was 16b0110000000110000, representing a fraction of 0.37574.

Figure 7.6: Locked Waveforms REFM and DIVCLK with ΔΣ Enabled

The phase noise of the internal reference REFM is shown in Figure 7.7. Notice that REFM is an irregular clock. The phase noise is strictly defined for periodic signals with a dominant carrier. For an irregular clock like REFM, which contains multiple repeating cycles, the equipment finds an average frequency from this patterned clock and calculates the phase noise relative to that carrier. This explains the carrier frequency of 51.32MHz. The phase noise of the test clock is shown in figure 7.8. The test clock is the VCO clock divided by 8. For comparison, the phase noise of the patterned clock from a signal generator is shown in Figure 7.9.

7.3 PLL IN CONVENTIONAL OPERATING MODE

In this section, we show various measurements for the PLL while putting it in the conventional mode of operation, in which a regular clock (square wave) is sent through

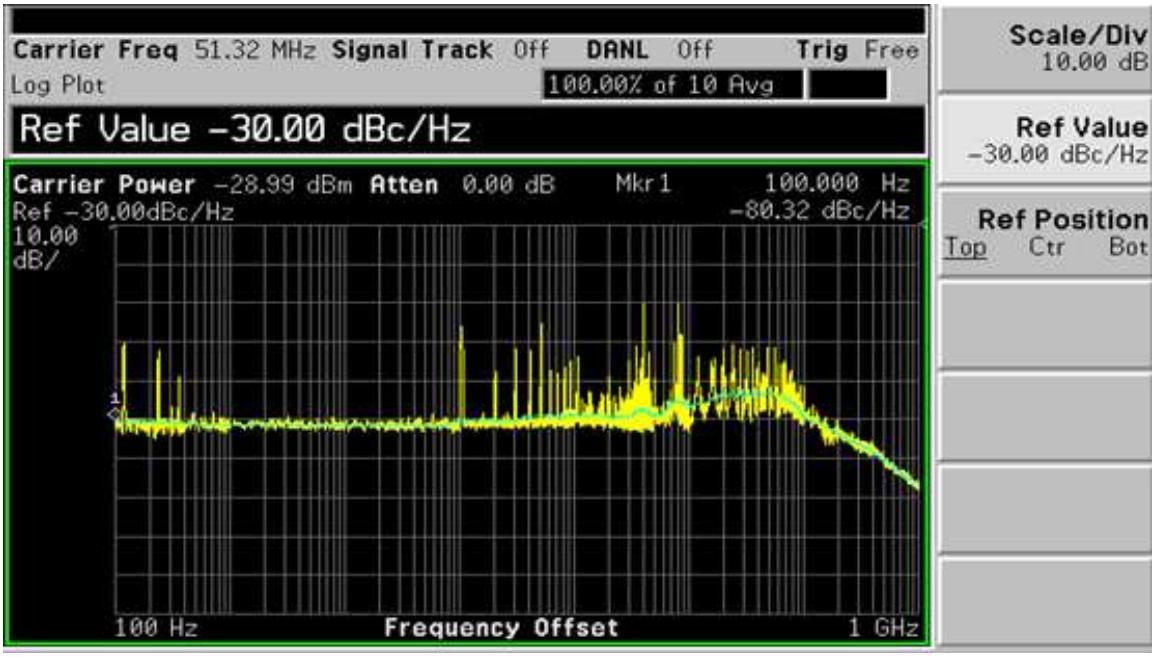


Figure 7.7: Phase Noise of the Patterned Internal Reference REFM

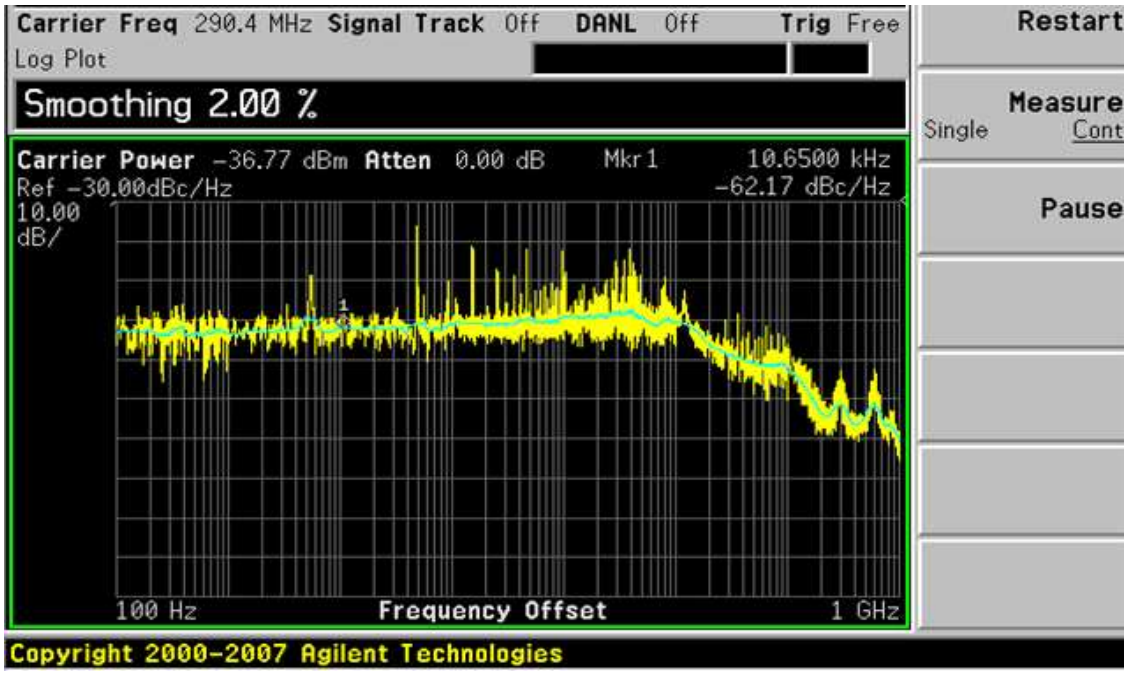


Figure 7.8: Phase Noise of the DIV8

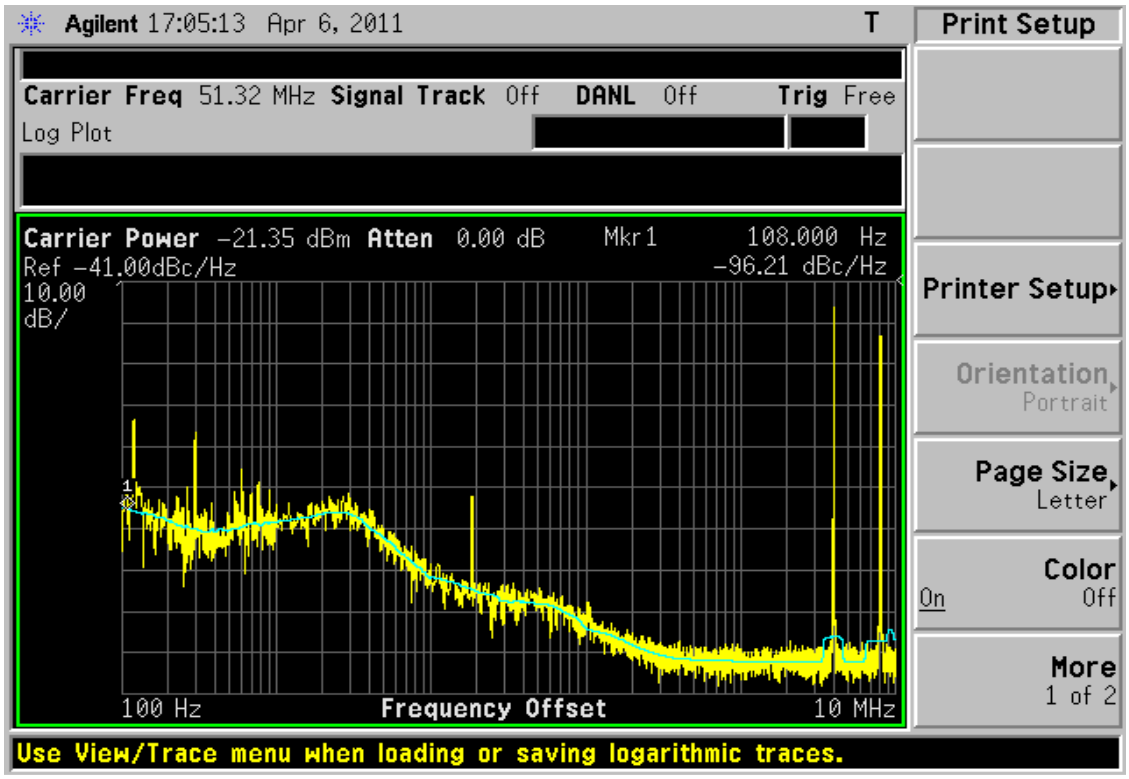
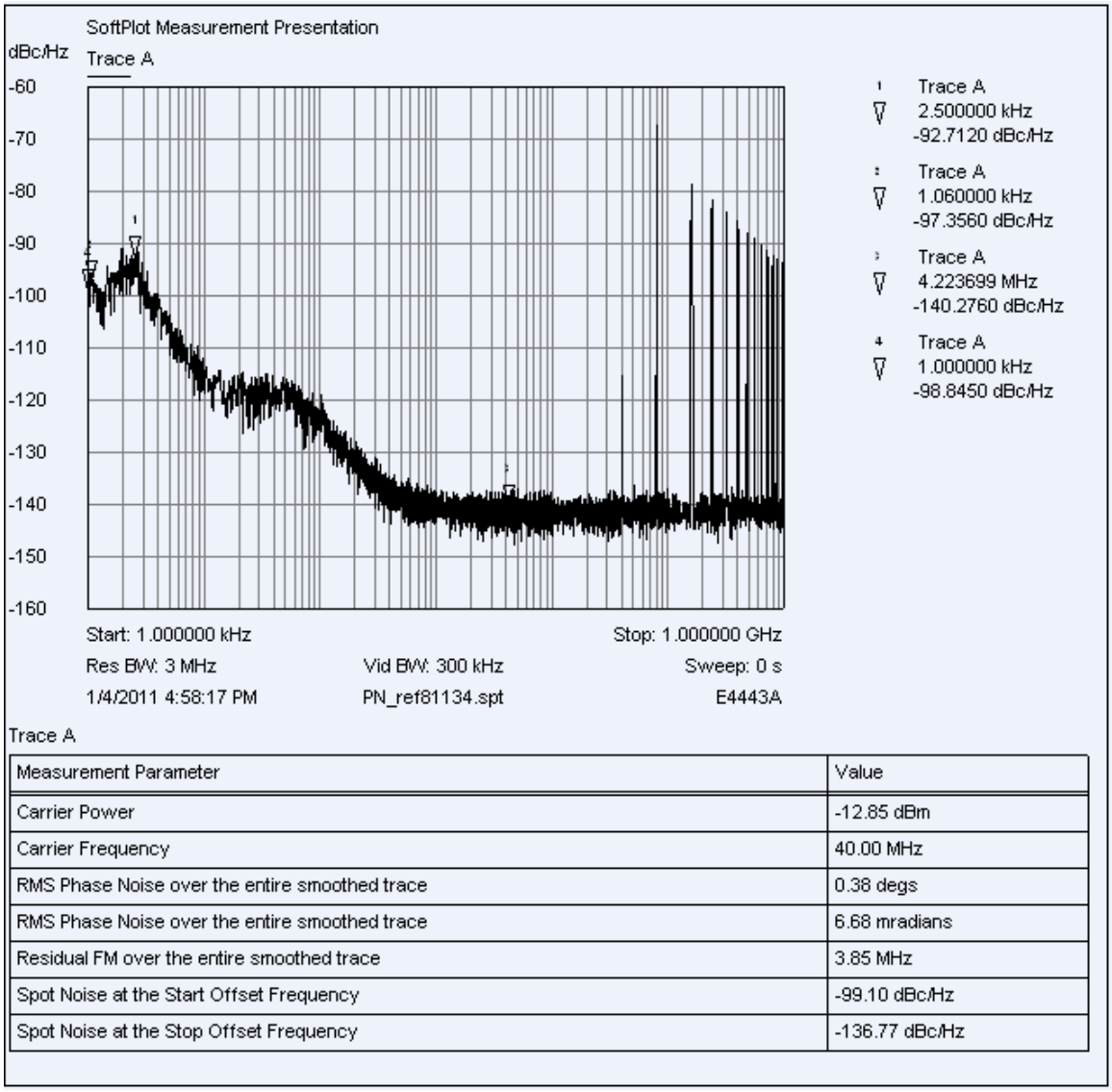


Figure 7.9: Phase Noise of the Patterned Clock from Agilent 81134

XO port bypassing the reference multiplier. The divide pattern is turned off and a fixed N ratio is used to control the divider. The ΔΣ can be enabled with a fractional ratio if desired. To fully characterize the loop, we used different input frequencies, and divide ratios. We measured the phase noise of XO, REFM, and DIV8. In this case REFM is an even clock, it is the same clock as input XO but going though internal buffers (however the offset buffers are bypassed). Figure 7.10 shows the phase noise of the reference from the source. It is a 40MHz clock from the Agilent 81134, with an integrated phase error of about 7 milli-radians. Figure 7.11 shows the phase noise of the same clock but internal to the chip. This clock is taken out through a test chip pad. It can be seen that the phase noise has degraded somewhat from the source, especially at high frequencies. There is up



Trace A

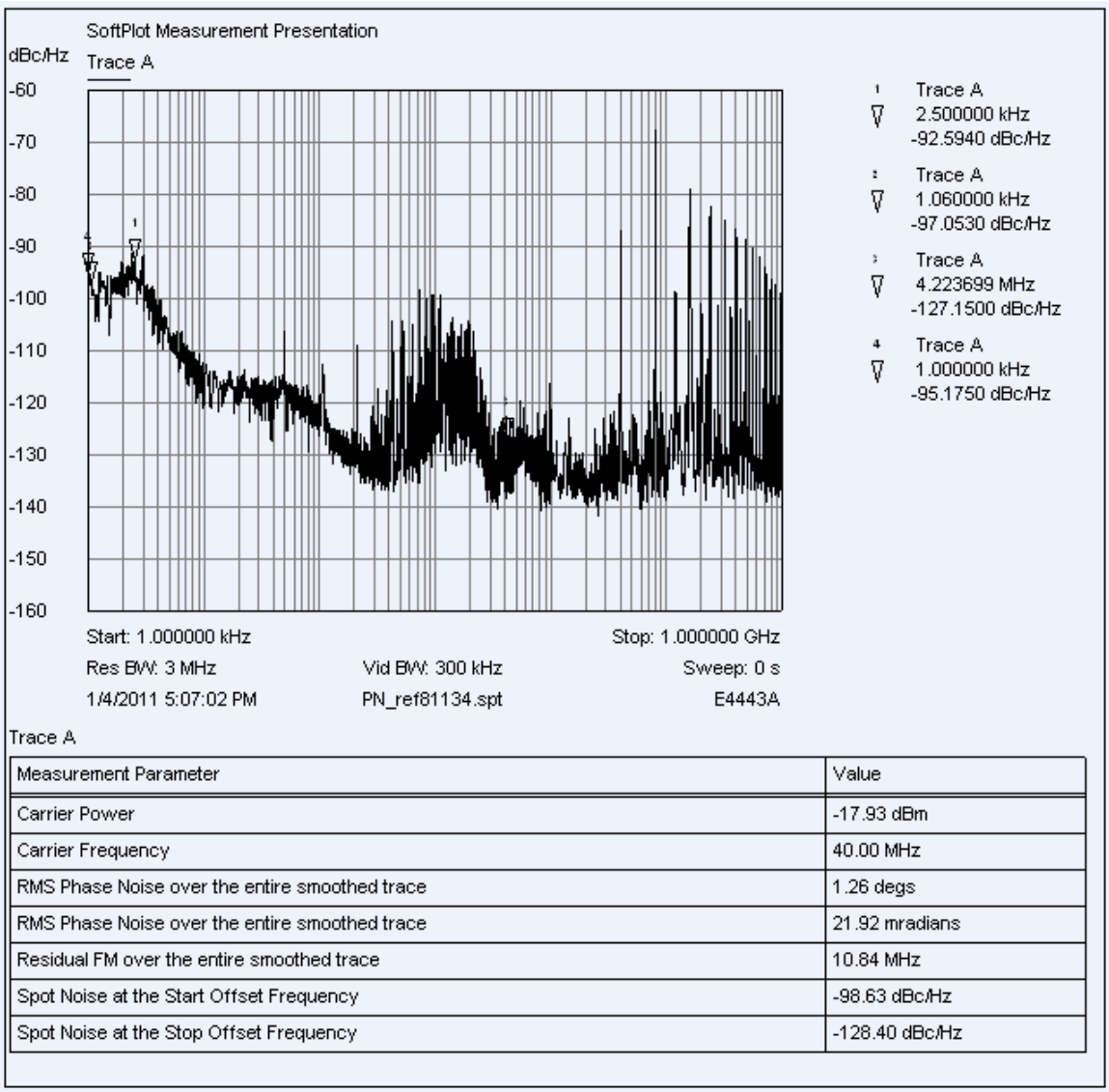
Measurement Parameter	Value
Carrier Power	-12.85 dBm
Carrier Frequency	40.00 MHz
RMS Phase Noise over the entire smoothed trace	0.38 degs
RMS Phase Noise over the entire smoothed trace	6.68 mradians
Residual FM over the entire smoothed trace	3.85 MHz
Spot Noise at the Start Offset Frequency	-99.10 dBc/Hz
Spot Noise at the Stop Offset Frequency	-136.77 dBc/Hz

Figure 7.10: Phase Noise of the Source (from Agilent 81134)

to 10dB increase in phase noise from 1M to 1GHz. The total integrated phase noise has increased from 7 milli-radians to 21 milli-radians. The source of this noise is analyzed in the Appendix A.

The phase noise for the DVI8 is shown in Figure 7.12. The divider ratio was set to 59, and the VCO frequency is measured to 295MHz. The spectrum of the divided clock is

shown in Figure 7.13. This is the feedback clock from the divider, which is locked to the input clock of 80MHz. The spectrum is shown in Figure 7.13 with the ΔΣ off, and in figure 7.14 with the ΔΣ on. Figure 7.15 shows the spectrum of the DIV8 clock.



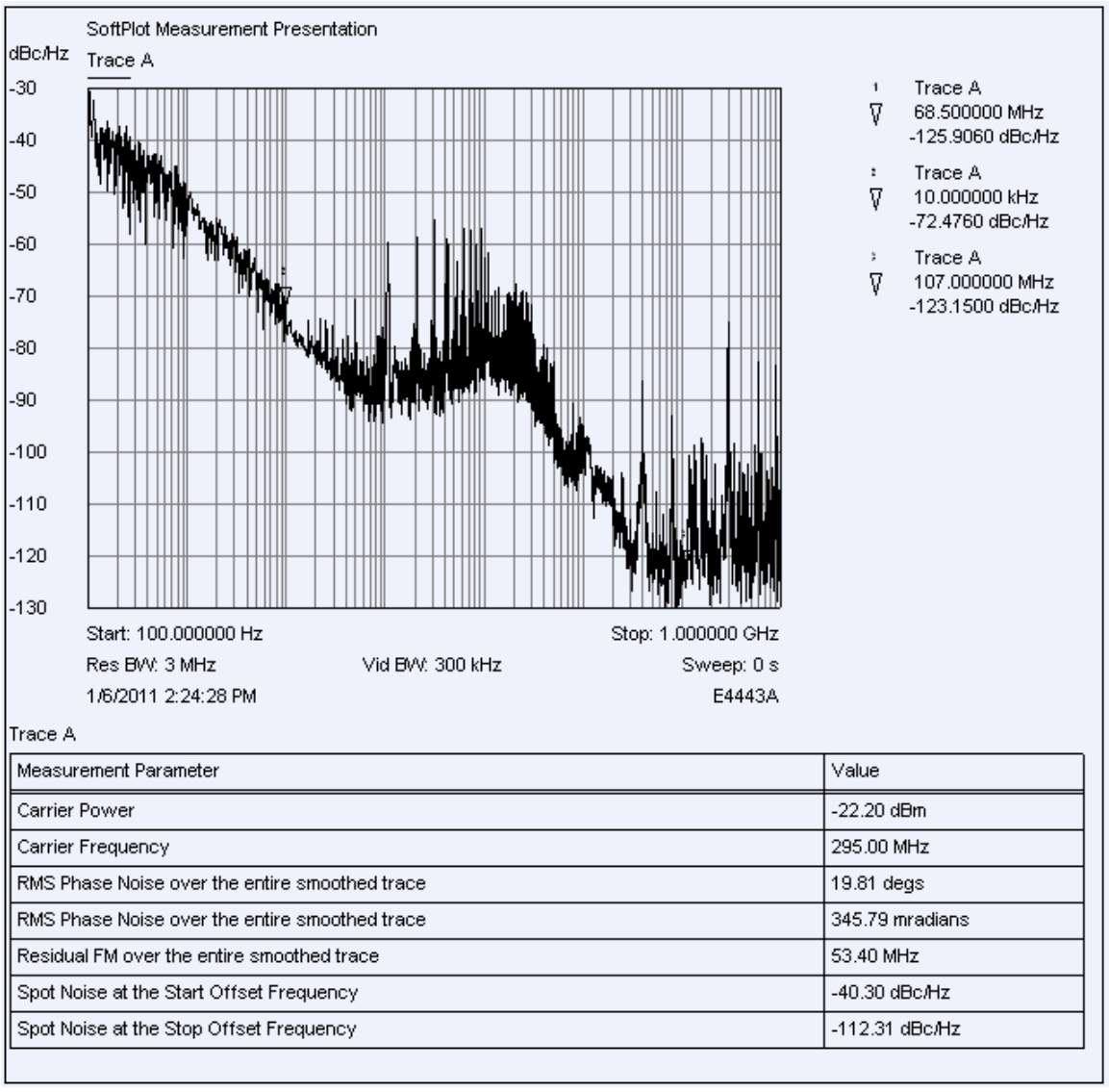
Trace A

Measurement Parameter	Value
Carrier Power	-17.93 dBm
Carrier Frequency	40.00 MHz
RMS Phase Noise over the entire smoothed trace	1.26 degs
RMS Phase Noise over the entire smoothed trace	21.92 mradians
Residual FM over the entire smoothed trace	10.84 MHz
Spot Noise at the Start Offset Frequency	-98.63 dBc/Hz
Spot Noise at the Stop Offset Frequency	-128.40 dBc/Hz

Figure 7.11: Phase Noise of the Reference after Internal Buffering

7.4 SUMMARY

Test chip measurement results are presented in this chapter. The measurements were taken under various operation modes of the PLL, both integer-N and fractional-N,



Trace A

Measurement Parameter	Value
Carrier Power	-22.20 dBm
Carrier Frequency	295.00 MHz
RMS Phase Noise over the entire smoothed trace	19.81 degs
RMS Phase Noise over the entire smoothed trace	345.79 mradians
Residual FM over the entire smoothed trace	53.40 MHz
Spot Noise at the Start Offset Frequency	-40.30 dBc/Hz
Spot Noise at the Stop Offset Frequency	-112.31 dBc/Hz

Figure 7.12 Phase Noise of the DIV8 clock.

with either a conventional reference, or with the proposed irregular reference. Under all conditions the feedback clock and reference clock are locked and match with full chip

level simulations. This proves the feasibility of the proposed architecture. We demonstrate for the fist time that a PLL can be locked with an irregular reference. This has never been published before. By using the proposed reference multiplier we obtained

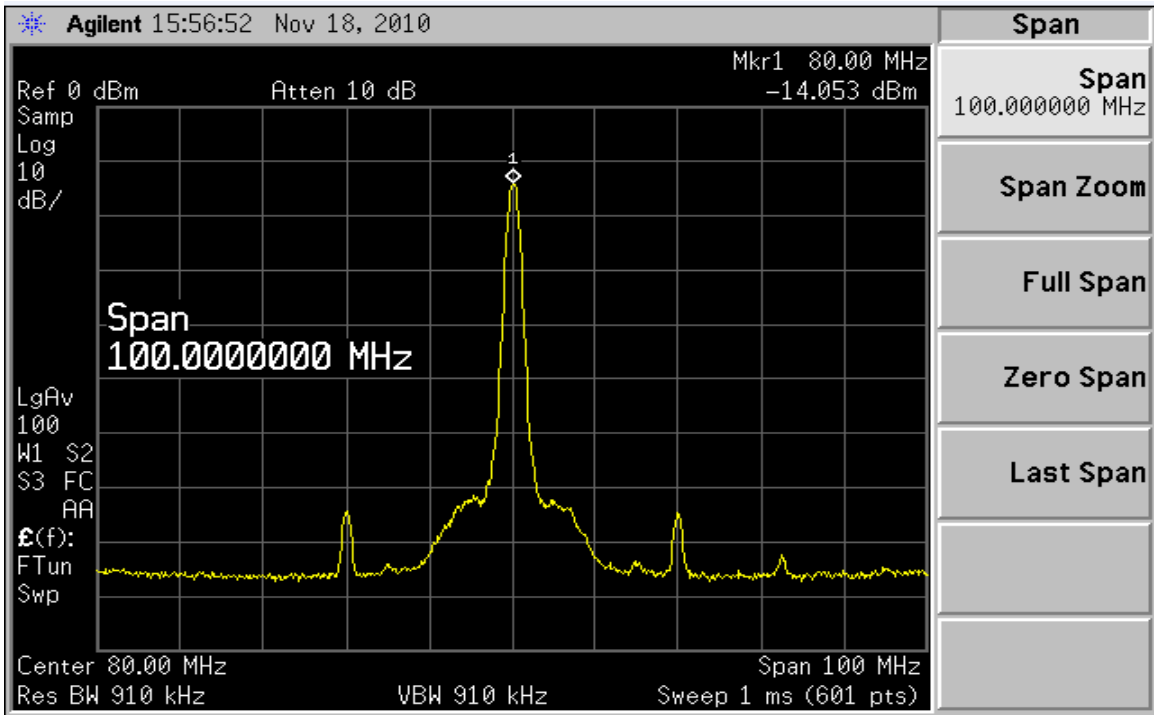


Figure 7:13 Spectrum of the DIVCLK with ΔΣ off

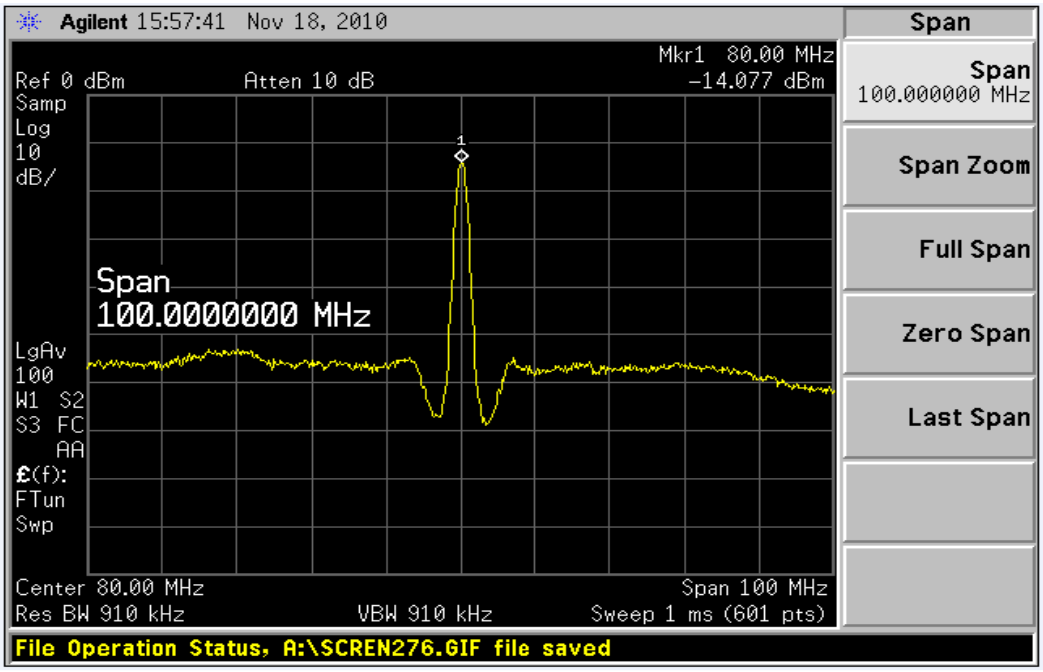


Figure 7.14: Spectrum of the DIVCLK with ΔΣ on

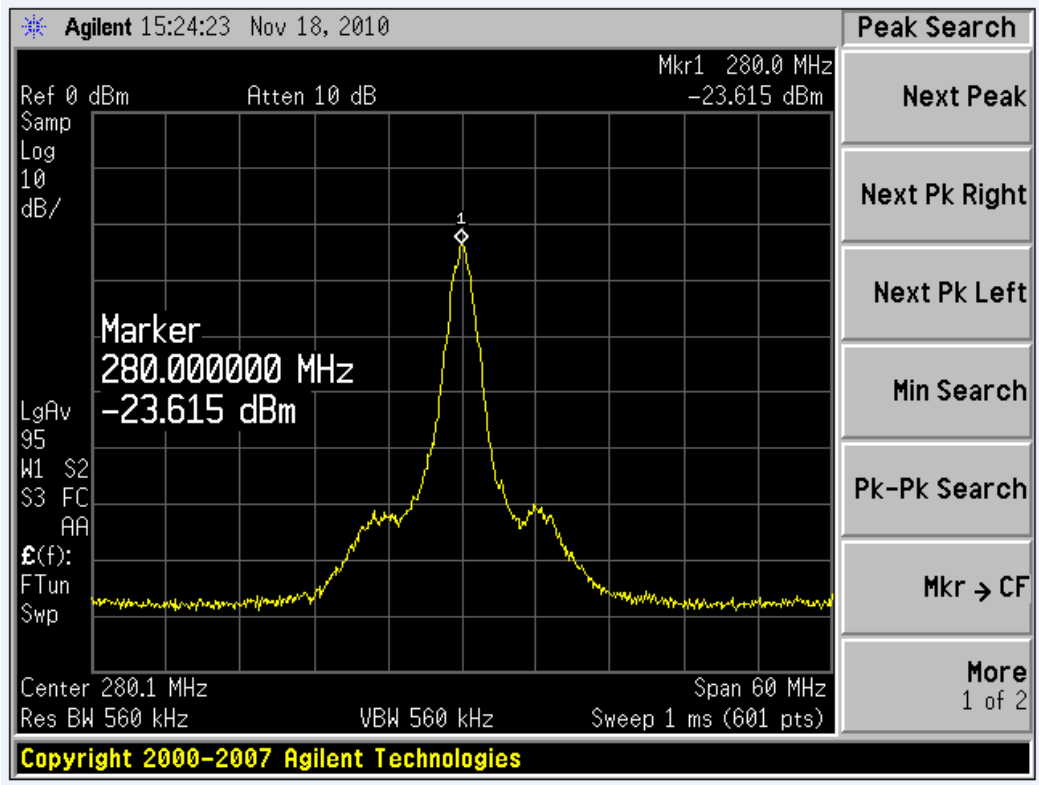


Figure 7.15: Spectrum of the DIV8

an internal reference which on average is 6x higher than the original reference. We have shown in simulations that a 6x higher reference provides a 24dB quantization noise reduction in a fractional-N PLL. This can also be traded off for a 6x increase in loop BW.

Chapter 8: Conclusions

8.1 CONTRIBUTIONS

The fractional-N PLL is a well known technique to synthesize fine resolution frequency sources from a fixed reference. Wide loop bandwidth is desirable for many applications where such a PLL is used. However a wide loop bandwidth reduces the effective oversampling ratio and makes quantization noise from the ΔΣ modulator a much bigger noise contributor. In this dissertation we presented the concept and implementation of a new PLL, where simple clock buffers with built-in offsets are used to extract six edges per reference cycle, effectively forming a reference multiplier. This allows for phase update to occur at a rate six times higher than the reference frequency. The higher update rate is particularly advantageous for quantization noise shaping in a ΔΣ controlled fractional-N PLL. A higher update rate increases the effective OSR, which allows most of the quantization noise to fall outside the PLL BW. Since the proposed reference multiplier utilizes the magnitude information from a sinusoidal reference to obtain the phases, the derived new clock edges tend to cluster around the zero-crossings and form an irregular clock. This presents a challenge in PLL lock acquisition. To the best of our knowledge the concept of using an irregular clock for phase locking has never been published before. In this work, the irregularity of the clock is taken into account in the divider by adding a cyclic divide pattern along with the ΔΣ control bits, this forces the loop to locally match the incoming patterns and achieve lock. Theoretically this new architecture allows for a 6x increase in loop BW or a 24dB improvement in phase noise. This approach can be easily extended to quadrature references as described in Section 8.2. In that case a 12x increase in BW or 32dB phase noise reduction can be achieved. Compared to DAC based phase noise cancellation approaches, where a high frequency

reference is needed to ease the matching requirements between the signal path and cancellation path, our approach can tolerate much lower references (for example 4MHz was used for this test chip). One potential issue associated with the proposed approach is the degraded spurious performance due to PVT variations, which lead to unintended mismatches between the irregular period and the divider pattern. A calibration scheme was invented to overcome this issue. In simulation, the calibration scheme was shown to lower the spurs down to inherent spurs level, which is insignificant compared to the total noise power. A test chip was taped out for proof of concept. Silicon measurements demonstrated the feasibility of the fundamental principles presented in this work.

8.2 FUTURE WORK

In this work we proposed a simple technique to produce a frequency multiplied reference and demonstrated for the fist time that the higher frequency, irregular clock can be used to lock a PLL. The variations among the irregular cycles alter the instantaneous loop gain which affects loop dynamics and stability. This presents a fundamental limitation for the proposed approach. This problem suggests a number of research directions that need to be pursued to improve the system.

The wide variations among the irregular cycles can be drastically reduced if a quadrature signal is used as the reference. A sine and a cosine signal complement each other in term of phase to voltage gain. As the phase to voltage gain for a sine reference decreases, the gain for a cosine reference increases, where addition clock edges can be created. The resulting reference is still an irregular clock but with much less variations. Quadrature signals can be created using a simple RC network poly phase filter [52]. In the simplest form an RC path provides a 45° phase shift, and a CR path provides a -45° phase shift producing a total of 90° offset. When a quadrature reference is used, the

proposed reference multiplier creates an internal reference which on average is 12x higher in frequency. This can be used for BW extension or traded off for phase noise reduction. A 12x increase in OSR would reduce quantization noise by 32dB. Since the variations among the irregular cycles are much reduced, this improves loop dynamics.

We have proposed two calibration schemes in Section 5.2 and 5.3. The first one calculates the total spur energy (in frequency domain) and uses it to control the calibration runs. The second one calculates the accumulated phase error (in time domain) and uses it to control the calibration runs. In both cases the calibration cycle finishes when either the spur energy or the accumulated phase error is within a pre-determined level. The first proposed calibration scheme requires a DSP engine to calculate the spur energy. A general purpose DSP engine is widely used in many SoCs. In the cases where such an engine is unavailable, a simplified DSP engine can be built using a recursive algorithm to calculate the discrete Fourier transform [53], [54]. Since this calibration scheme only calls for a few discrete frequency bins to estimate the total spurious energy, a full-blown DSP engine is no needed. However, this approach requires accurate jitter measurement on the output clock, which becomes difficult when a good reference is not available. The second proposed calibration scheme is an improvement over the first approach. In this approach only the time domain accumulated phase error is used, neither an accurate frequency reference nor a DSP engine is required. In the case of a digital phase detector, a simple running average can be used for calibration.

As described in the Appendix, an error in the output buffer introduced excessive coupling on to the reference and degraded the overall performance. This can be easily corrected in a silicon revision.

Appendix A Issues in Silicon

We have established the feasibility of the proposed approach through silicon measurements on REFM and DIVCLK. It proves that an irregular clock derived from a sinusoidal reference can be used to achieve phase locking by imposing a cyclic divide pattern to locally match the incoming reference pattern. However there are two issues in silicon that need to be addressed. We found that the test clock which is the VCO clock divide by 8 is not accurate for all the cases tested. The other issue is the excessive noise coupling on to the reference. We take a closer look at these issues in this appendix.

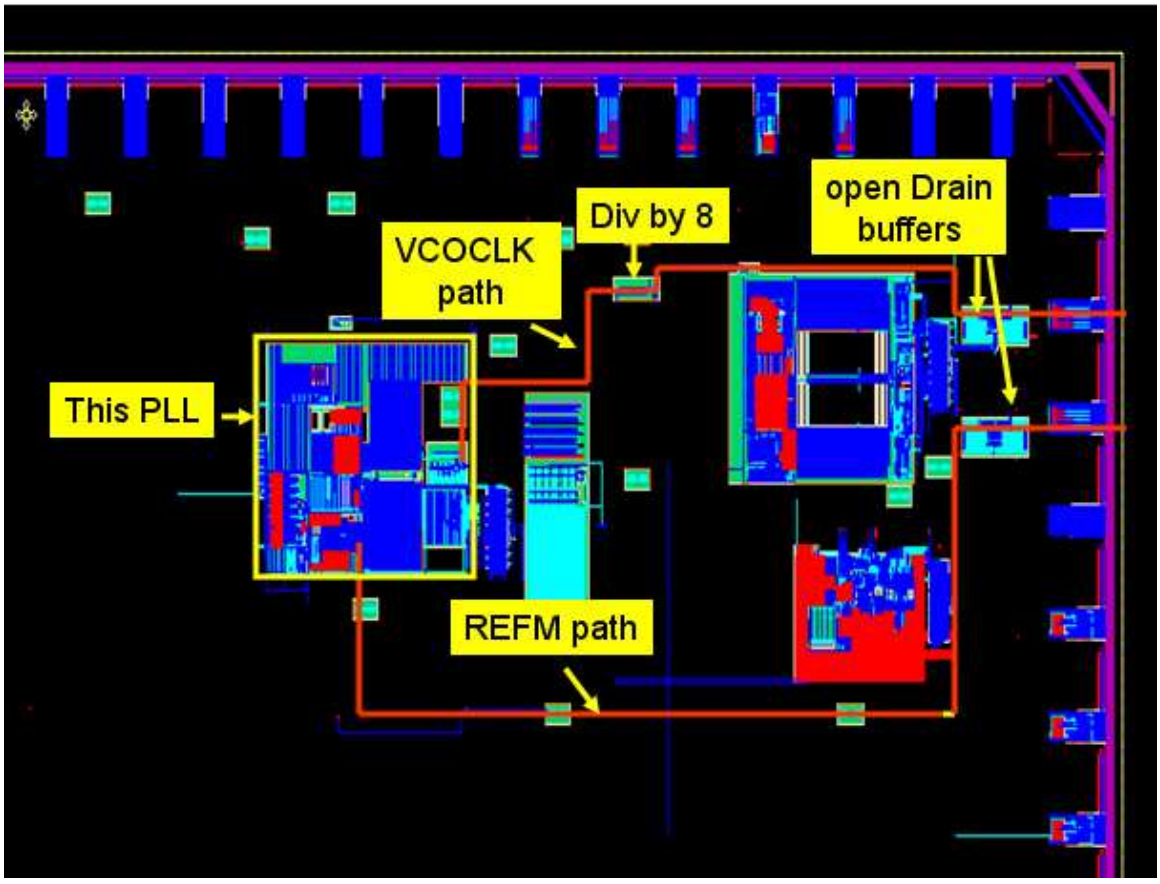


Figure A.1: Test chip Layout View for Clock Routes

The test chip layout is shown in Figure A.1. The PLL for this work is located to the left of the figure. The output clock from VCO (at 2.4 GHz) is routed in MET4 and MET5 for about 500um before it reaches the first set of buffers, it is then routed another 500um in MET5 and METTOP before feeds into the final divider of divide by 8. This long route presents a heavy load to the GHz clock with a lumped paracitic pole for the initial 500um is roughly 3.7GHz. The divide-by-8 circuitry is implemented using a three-stage divide-by-2 CML logic. CML logic is known to be very sensitive to the DC levels of the input clock and the rise and fall times of the clock edges. Based on silicon measurements, it is likely a miscount occurred in the final divider of divide by 8, this would add an offset to the measured frequency. This divider is placed at the test chip level outside the PLL. Since the VCO is located at the upper left corner, the route is unnecessarily long. A better placement for this divider is the space outside the PLL immediately above the VCO to prevent a long rout for the GHz clock. This correction will be implemented in the next silicon revision.

The second issue regards to noise coupling on the internal reference. The phase noise of XO, REFM and DIV8 are re-plotted in Figure A.2. Compared to the phase noise of the XO, the REFM shows excessive noise beyond 200 kHz. We determined that the 15dB noise gain near 1MHz region is due to the activities from the FPGA box, it is also seen on the output clock, DIV8. This problem can be removed by designing a default setting that does not require FPGA controls. In addition, there are excessive high frequency noise and tones on REFM. The source of these contaminations if from the open-drain buffer. The open-drain buffer used by the REFM also shares the common supply with another open-drain buffer, which is used to drive DIV8 clock off chip. The open-drain buffers used here draw significant current from supply. It has been seen on the bench that there is a strong cross-talk between the two output buffers. The problem can

be eliminated in silicon revision by removing one of the open-drain buffers used for REFM and instead used a quiet analog pin to this signal.

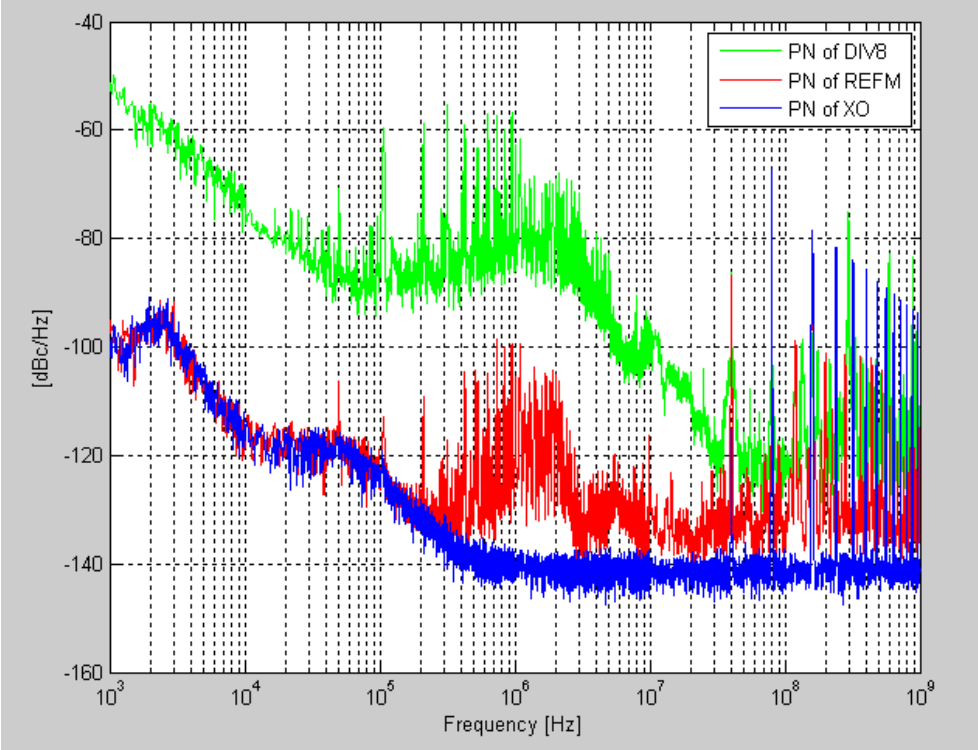


Figure A.2: Phase Noise Illustrating Reference Corruption

Appendix B Synthesizable Digital RTL Models

```
module dsm2_pattern (
	// Inputs
	vin,
	clk,
	rstb,
	en_patt,
	en_IQ,
	en_dsm,
	Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,
	// Outputs
	yout );

//////Inputs
input [15:0] vin;
input	clk;
input	rstb;
input		en_patt, en_IQ, en_dsm;
input [7:0] Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11;//divide pattern

/////// Outputs
output [7:0] yout;

reg [7:0] yout;
wire [3:0] count;
wire [7:0] Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11;
wire [7:0] mux_out;
wire [7:0] pattern;
wire [2:0] pctrl;
wire [4:0] dctrl;
wire [2:0] dout;

count count1(.Q(count), .clk(clk), .rstb(rstb), .en_patt(en_patt), .enIQ(en_IQ));

mux6to1 mux1 (.out(mux_out), .count(count),
.Q0(Q0),.Q1(Q1),.Q2(Q2),.Q3(Q3),.Q4(Q4),.Q5(Q5),.Q6(Q6),.Q7(Q7),.Q8(Q8),.Q9(Q9),.Q10(Q10),.Q11(Q11));

dsm2 dsm2_core (.vin(vin),.clk(clk),.rstb(en_dsm&rstb), .dout(dout));

assign pattern=mux_out+{dout[2],dout[2],dout[2],dout[2],dout[2],dout[2:0]};
assign pctrl = pattern[2:0];
assign dctrl = pattern[7:3]+5'b11110;

always @(posedge clk or negedge rstb)
begin
		if(~rstb)
		  yout<=8'b11100000;
		else
		  yout<={dctrl,pctrl};
end

endmodule
```

```
module mux6to1(out, count, Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11);

 output[7:0] out;
 input [7:0] Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11;
 input [3:0] count;
 reg [7:0] out;

 always @ (count or Q0 or Q1 or Q2 or Q3 or Q4 or Q5 or Q6 or Q7 or Q8 or Q9 or Q10 or Q11)

   case (count)
     4'b0000: out=Q0;
     4'b0001: out=Q1;
     4'b0010: out=Q2;
     4'b0011: out=Q3;
     4'b0100: out=Q4;
     4'b0101: out=Q5;
     4'b0110: out=Q6;
     4'b0111: out=Q7;
     4'b1000: out=Q8;
     4'b1001: out=Q9;
     4'b1010: out=Q10;
     4'b1011: out=Q11;

     default: out=Q0;
   endcase
endmodule

module count(Q, clk, rstb, en_patt, enIQ);

output[3:0] Q;
input clk, rstb, en_patt, enIQ;
reg [3:0] Q;

always @ (posedge clk or negedge rstb)
begin
  if (~rstb) Q<=4'b0000;
  else begin
            if (enIQ)
       Q<=(Q+1) % 4'b1100;
            else if (en_patt)
            Q<=(Q+1) % 4'b0110;
            else
            Q<=4'b0000;
      end
end
endmodule
```

```
module dsm2 ( vin, clk, rstb, dout );

input [15:0] vin;
input        clk;
input        rstb;
output [2:0] dout;

wire [17:0] v1;
wire [17:0] v11;
wire [17:0] d_s1;
wire [18:0] d_s2;
wire [18:0] v2;
wire [18:0] v22tmp;
wire [17:0] v22;
wire [17:0] vsum;
wire [2:0] d;
wire xr;
wire [15:0] xr_signed;
wire [15:0] vin_xr;
reg [17:0] v11_q;
reg [17:0] v22_q;
reg [2:0] dout;

lfsr lfsr_inst (.s_out(xr), .clk(clk), .rstb(rstb));
assign xr_signed = (xr)? 16'b0000000000000001:16'b1111111111111111;
assign vin_xr =vin + xr_signed;
assign v1 = {vin_xr[15], vin_xr[15], vin_xr} + v11_q;
assign v11 = v1 + ~d_s1 + 1'b1;
assign v2 =  {v11_q[17],v11_q} + {v22_q[17],v22_q};
assign v22tmp = v2 + ~d_s2 + 1'b1;
assign v22 = v22tmp[17:0];
assign vsum = v22_q + 18'b000100000000000000;
assign d = vsum[17:15];
assign d_s1 = d << 15;
assign d_s2 = d_s1 << 1;

always @ (posedge clk or negedge rstb)
begin
       if (~rstb)
         begin
              v11_q <= 18'b0;
              v22_q <= 18'b0;
         end
       else
         begin
              v11_q <= v11;
              v22_q <= v22;
              dout <= d;
         end
end

endmodule
```

```
read_lib GS70_W_125_0.99_0.99_CORE.lib
set link_path "* GS70_W_125_0.99_0.99_CORE.db"
set target_library GS70_W_125_0.99_0.99_CORE.db
analyze -f verilog ./lfsr.v
analyze -f verilog ./count.v
analyze -f verilog ./mux6to1.v
analyze -f verilog ./dsm2.v
analyze -f verilog ./dsm2_pattern.v
elaborate lfsr -lib work
elaborate count -lib work
elaborate mux6to1 -lib work
elaborate dsm2 -lib work
elaborate dsm2_pattern -lib work
set current_design dsm2_pattern
link
uniquify
ungroup -flatten -all
set_load 10 [all_outputs]
set_max_area 0
create_clock -name clk -period 1800 [ get_ports {clk}]
set_input_delay -max 400 -clock clk [ get_ports {Q*}]
set_input_delay -min 0 -clock clk [get_ports {Q*}]
set_output_delay -max 1000 -clock clk [ get_ports {yout}]
set_output_delay -min 0 -clock clk [ get_ports {yout}]
#set_dont_use "GS70_W_125_0.99_0.99_CORE.db/enor2_f0p5_hd GS70_W_125_0.99_0.99_CORE.db/exor2_f0p5_hd"
source synth_cells_dontuse.tcl
compile -map_effort high
report_area
report_timing
report_reference
write -f verilog dsm2_pattern -output dsm_post_synth.v -hierarchy
```

Bibliography

[1] Electronics tutorials about Quartz Crystal Oscillator "http://www.electronics-tutorials.ws/oscillator/crystal.html

[2] W. F. Egan, *Frequency Synthesis by Phase Lock*, 2nd ed. New York: Wiley Interscience, 2000.

[3] B. De Muer, M.S.J. Steyaert, *CMOS Fractional-N Synthesizers*, Kluwer Academic Publishers, 2003

[4] S. Pamarti, L. Jansson, and I. Galton, "A wideband 2.4 GHz delta-sigma fractional-N PLL with 1-Mb/s in-loop modulation," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 49–62, Jan. 2004.

[5] M. Gupta and B. Song, "A 1.8-GHz Spur-Cancelled Fractional-N Frequency Synthesizer With LMS-Based DAC Gain Calibration" *IEEE Journal of Solid State Circuits*, vol. 41, no. 12, December 2006.

[6] A. Swaminathan, K. Wang, I.Galton," A Wide-Bandwidth 2.4GHz ISM-Band Fractional-N PLL with Adaptive Phase-Noise Cancellation" in *IEEE International Solid-State Circuits Conference Dig. Tech. Papers*, pp 302-304, Feb. 2007.

[7] S. E.Meninger, and M.H.Perrott, "A 1 MHz Bandwidth, 3.6 GHz, 0.18um CMOS Fractional-N Synthesizer Utilizing a Hybrid PFD/DAC Structure for Reduced Broadband Phase Noise." *IEEE Journal of Solid State Circuits*, vol. 41, no. 4, April 2006.

[8] E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700 kHz bandwidth fractional synthesizer with spurs compensation and linearization techniques for WCDMA applications," *IEEE Journal Solid-State Circuits*, vol. 39, no. 9, pp. 1446–1454, Sep. 2004.

[9] B. De Muer and M. S. J. Steyaert, "A CMOS monolithic delta-sigma controlled fractional-N frequency synthesizer for DCS-1800," *IEEE Journal Solid-State Circuits*, vol. 37, no. 7, pp. 835–844, Jul. 2002.

[10] W. Rhee and A. Ali, "An on-chip compensation technique in fractional-N frequency synthesis," in *Proceedings of IEEE International Symposium on Circuits and System*s, pp. 363–366, 1999.

[11] A. Thomsen et al, "A Fractional-N PLL for SONET-Quality Clock-Synthesis Applications" in *IEEE International Solid-State Circuits Conference Dig. Tech. Papers*, Feb. 2007, pp 58-60.

[12] Y. Dufour, "Method and apparatus for performing fractional division charge compensation in a frequency synthesizer," U.S. Patent 6,130,561, Oct. 10, 2000.

[13] C. Heng and B. Song, "A 1.8-GHz CMOS fractional-N frequency synthesizer with randomized multiphase VCO," *IEEE Journal Solid-State Circuits*, vol. 38, pp. 848–854, June 2003.

[14] C. Park, O. Kim, and B. Kim, "A 1.8 GHz self-calibrated phase-locked loop with precise I Q matching," *IEEE Journal Solid-State Circuits*, vol. 36, pp. 777–783, May 2001.

[15] K. Lee *et al.*, "A single-chip 2.4-GHz direct-conversion CMOS receiver for wireless local loop using multiphase reduced frequency conversion technique," *IEEE Journal Solid-State Circuits*, vol. 36, pp. 800–809, May 2001.

[16] Temporiti, E.; Weltin-Wu, C.; Baldi, D.; Cusmai, M.; Svelto, F., "A 3.5 GHz Wideband ADPLL With Fractional Spur Suppression Through TDC Dithering and Feedforward Compensation", *IEEE Journal Solid-State Circuits*, vol. 45, pp. 2723–2736, Dec. 2001.

[17] R. B. Staszewski *et al.*, "All digital PLL and transmitters for mobile phones," *IEEE Journal Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.

[18] C.-M. Hsu, M. Z. Straayer, and M. H. Perrott, "A low-noise wide-BW 3.6-GHz digital ΔΣ Fractional-N Frequency Synthesizer with a Noise Shaping Time-to-Digital Converter and Quantization Noise Cancellation", *IEEE Journal Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec. 2008.

[19] M. Lee, M. E. Heidari, and A. A. Abidi, "A low-noise wideband digital phase-locked loop based on a coarse-fine time-to-digital converter with sub-picosecond resolution," *IEEE Journal Solid-State Circuits*, vol. 44, no. 10, pp. 2808–2816, Oct. 2009.

[20] M. Zanuso, P. Madoglio, S. Levantino, C. Samori, and A. Lacaita, "A 3 MHz-BW 3.6 GHz digital fractional- PLL with sub-gate-delay TDC, phase-interpolation divider and digital mismatch cancellation," in *IEEE International Solid-State Circuits Conference Dig. Tech. Papers*, pp. 476–477, Feb. 2010.

[21] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York: McGraw Hill, 2001, pp. 562–566.

[22] A. Hajimiri and T. Lee, "A general theory of phase noise in electrical oscillators," *IEEE Journal Solid-State Circuits*, vol. 33, pp. 179–194, Feb. 1998.

[23] Thomas Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press; 2nd ed., 2003

[24] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation* IEEE Press, 1997.

[25] G. Chien and P. Gray, "A 900MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," in *IEEE International Solid-State Circuits Conference Dig. Tech. Papers*, Feb. 2000, pp. 202–203.

[26] R. Farjad-Rad, W. Dally, Ng Hiok-Tiaq et. al. "A Low-Power Multiplying DLL for Low-Jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips," *IEEE Journal Solid-State Circuits*, vol. 37, no. 12, pp. 1804-1812, Dec. 2002.

[27] S. Ye and I. Galton, "Techniques for Phase Noise Suppression in Recirculating DLLs," *IEEE Journal Solid-State Circuits*, vol. 39, no. 8, pp. 1222-1230, Aug. 2004.

[28] R. B. Staszewski et al, "All-Digital TX Frequency Synthesizer and Discrete-Time Receiver for Bluetooth Radio in 130-nm CMOS" *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2278– 2291, Dec. 2004.

[29] R.E. Best, *Phase-Locked Loops*, McGraw-Hill Professional; 5th ed., 2003.

[30] K. Nagaraj, *APLL Analysis*, TI internal communications, 2006

[31] Silicon Labs AN256.

[32] Lecture notes on Radio Frequency Integrated Circuits, J. Dąbrowski, Linköping University.

[33] Lecture note on Mixed Signal Design, E. Swanson, UT Austin.

[34] Lecture note on Clock and Data Recovery Circuits, James S. Kenney, Georgia Tech.

[35] D.C. Lee, "Analysis of jitter in phase-locked loops"**,** *IEEE Transaction on Circuits and Systems II*, vol. 49, no 11, pp. 704-711, 2002.

[36] M. H. Perrott, T. L. Tewksbury III, and C. G. Sodini, "A 27-mW CMOS fractional-N synthesizer using digital compensation for 2.5-Mb/s GFSK modulation," IEEE Journal Solid-State Circuits, vol. 32, pp. 2048–2059, Dec.1997.

[37] Dick Bension, Analog/Mixed Signal Examples, The Mathworks.

[38] J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables.* Classics in Applied Mathematics, SIAM, 2000.

[39] J. Craninckx, M. Steyaert, *Wireless CMOS Frequency Synthesizer Design*, Kluwer Academic Publishers, 1998.

[40] R. Hogervorst, J. P. Tero, R.G. H. Eschauzier, and J. H. Huijsing, "**A** Compact Power-Efficient 3 V CMOS Rail-to-Rail Input/Output Operational Amplifier for VLSI Cell Libraries" *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1505-1513, Dec. 1994.

[41] J. H. Huijsing, R. Hogervorst, and K. de Langen "Low-Power, Low-Voltage VLSI Operational Amplifier Cells, *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 42, no.11, pp. 841-852, Nov. 1995

[42] C.Park, B. Kim, "A Low-Noise, 900-MHz VCO in 0.6- m CMOS", *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, pp. 586-592, May 1999.

[43] C. S. Vaucher and D. Kasperkovitz, "A wide-band tuning system for fully integrated satellite receivers," *IEEE Journal Solid-State Circuits*, vol. 33, pp. 987–998, July 1998.

[44] S. Yu, P. Kinget "A 0.65V 2.5GHz Fractional-N Frequency Synthesizer in 90nm CMOS", *IEEE International Solid-State Circuits Conference*, Feb. 2007.

[45] C. Hung, Ph.D. dissertation "Investigation of a Multi-GHz Single-chip CMOS PLL Frequency Synthesizer for Wireless Applications", 2000.

[46] Y. Ding, K.K. O, A 21-GHz 8-Modulus Prescaler and a 20-GHz Phase-Locked Loop Fabricated in 130-nm CMOS, *IEEE Journal of Solid-State Circuits*, vol. 42, no. 6, pp. 1240-1250, June 2007

[47] NanoSim Reference Guide, Synopsys, TLD-2001.06, June 2001

[48] W. Rhee, "Design of high-performance CMOS charge pumps in phase locked loops," in *Proceedings IEEE International Symposium Circuits and Systems*, pp. 545–548, 1999.

[49] A. Abidi, R.G. Meyer, Noise in Relaxation Oscillators, *IEEE Journal of Solid-State Circuits*, vol. 18, no. 6, pp. 794-802, 1983.

[50] B. Razavi, "A study of phase noise in CMOS oscillators," IEEE Journal Solid-State Circuits, vol. 31, pp. 331–343, Mar. 1996.

[51] Phase Noise Guide, Agilent Technologies PSA and ESA-E Series Spectrum Analyzers Manual.

[52] J. Crols, M. Steyaert, "CMOS Wireless Transceiver Design", Kluwer Academic Publishers, 1997

[53] S. Lai, S. Lei, C. Chang, C. Lin, C. Luo, "Low Computational Complexity, Low Power, and Low Area Design for the Implementation of Recursive DFT and IDFT Algorithms", *IEEE Transactions on Circuits and Systems-II*: Express Briefs, vol. 56, no. 12, pp. 921-925, Dec. 2009

[54] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American Math Monthly*, vol. 65, pp. 34–35, Jan. 1958.

[55] X. Pu, A. Thomsen, "Fractional-N Frequency Synthesizer- a Novel Approach" *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pp 135-138, 2006.

[56] X. Pu, A. Thomsen, J. Abraham, "Improving Bandwidth while Managing Phase Noise and Spurs in Fractional-N PLL", *IEEE Computer Society Annual Symposium on VLSI*, 2008.

[57] X. Pu, J. Abraham, A. Thomsen and K. Nagaraj, "A Bandwidth-enhanced Fractional-N PLL through Reference Multiplication", *IEEE Transactions on Circuits and Systems-II*, submitted for publication.

[58] X. Pu, J. Abraham, A. Thomsen and K. Nagaraj, "A Novel Fractional-N PLL Based on a Simple Reference Multiplier", *IEEE International Midwest Symposium on Circuits and Systems*, accepted for publication.

[59] X. Pu, A. Kumar, S. Goldman and K. Nagaraj, "Fully Integrated 100 MHz-400 MHz Analog PLL Working from a 32 KHz Reference Input for Wireless Connectivity and Broadcast Applications", *IEEE Transactions on Circuits and Systems-II*, submitted for publication.

Permanent email : pu@ti.com

This dissertation was typed by Xiao Pu.